Electrical Load Forecasting Using Adaptive Neuro-Fuzzy Inference System

Doctoral Thesis

to be awarded the degree

Doctor of Engineering (Dr.-Ing.)

Submitted by

Ali Hashemifarzad, M.Sc.

Approved by the Faculty of Energy and Economic Sciences

Clausthal University of Technology

Date of Oral Examination

15.05.2019

Dean

Prof. Dr. rer. nat. habil. Bernd Lehmann

Chairperson of the Board of Examiners

Prof. Dr.-Ing. Norbert Meyer

Chief Reviewer

Prof. Dr.-Ing. Martin Faulstich

Reviewer

Prof. Dr. rer. nat. Christian Siemers

Declaration of Authorship

I hereby declare that this dissertation and the work presented in it are my own and has been generated by me as the result of my own original research, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted and all sources of information are always given. This doctoral thesis has not been accepted in any previous application for a degree.

…………………………………….

Ali Hashemifarzad

Clausthal-Zellerfeld, 22. October 2018

Acknowledgments

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Dr.-Ing. Martin Faulstich, who has supported me throughout this research with his patience and knowledge whilst allowing me the room to work in my own way.

I would like to thank my second supervisor Prof. Dr. rer. Nat. Christian Siemers, whose advices were the key to this successful research.

I would like to express the deepest appreciation to Dr.-Ing. Jens zum Hingst and Dr.-Ing. Werner Siemers for the wonderful working environment and their supports through this whole time. Without their guidance and persistent help, this dissertation and unforgettable experience would not have been possible.

The most special thanks goes to my loved ones: my dear mom, Soheila, my dad, Ahmad, my sister and brother; Tabassom and Reza, you gave me your unconditional support and love through all this long process and my life in general. I will be grateful forever for your supports.

Last but not the least, I would like to thank all my friends for accepting nothing less than excellence from me. You motivated me day by day and stayed beside me through thick and thin.

Ali Hashemifarzad

October 2018

Abstract

The importance of a precise load forecasting in energy system designing has been known since many years ago. To achieve an efficient, secure and economically optimal operation in an energy systems, it is necessary to analyze the power system operation in time and perform fast reactions to changes in electrical load. These are highly dependent on an accurate short-term load forecasting.

One the other hand, the long-term load prediction is also of great importance for energy suppliers, independent system operators, financial institutes and other parties involved in power generation, transmission, distribution and market. For the energy suppliers, a timely implementation of long-term load forecasting helps the system to improve the network stability and reduce the equipment failures and power outages and ensures a reliable energy supply.

This dissertation have tried to propose a new approach toward the problem of load forecasting to provide more accurate results with a lesser error rate compared to the current methods. For this, a methodology based on time series analysis and the *Chaos* and *Concept Drift* theories is introduced, in which the *Artificial Neuro Fuzzy Inference System (ANFIS)* plays the main role in training and testing the model.

As a case study, the electrical load of the Clausthaler Umwelttechnik Forschungszentrum (CUTEC) Institute from the year 2014 was selected. These data were analyzed by *mutual information function* to find the similarities and repetitions in it. This was inspired by the concept drift theory as a kind of anomaly existence in time series, which can help to classify the input data based on the similarities and prepare them for a systematic feeding in the training process. Afterwards, the existence of chaos in the time series was investigated by applying three different methods, namely, *Embedding dimension*, *Grassberger-Procaccia correlation dimension* and *Lyapunov exponents*. A chaotic time series is based on a deterministic system, i.e. it contains enough information to be predicted by the method proposed in this dissertation.

The training and testing process was done in MATLAB by using ANFIS and the forecasting was done for three different time horizon: Weekly, monthly and seasonal. The mean absolute percentage error (MAPE), as one of the most important indices in load forecasting, in all time horizons are between 2.1% and 2.6%, which are significantly lower than most reported similar cases in studies with almost the same datasets and prediction horizon, confirming that the

proposed method shows a more efficient, time-saving, and more accurate approach to the problem of electrical load prediction.

In addition to that, this research includes a simulation model for a decentralized energy system (DES) implemented in MATLAB and SIMULILNK. This model includes different units such as combined heat and power generation units (CHP), photovoltaics, solar thermal panels, boiler and storage units. The user can add the electrical and thermal load as a time series via a user interface and define the priority and additional details of each unit. The model provides information about the generation plan of each module and storage level, which can also be exported as an Excel file. In addition, various graphic representations such as generation and load vs. consumption diagrams can be provided.

This model has been validated with the Energy Park at the CUTEC Institute. The simulation results for 2014 were then compared with the results of the same simulation in EnergyPro, a commercial energy analysis software. The results were in accordance to each other with a negligible difference of less than 1%.

Kurzfassung

Die Kenntnis über den zukünftigen Bedarf an elektrischer Energie spielt eine entscheidende Rolle bei der Gestaltung der Energiesysteme von der Erzeugung bis zu der Nutzung. Die Lastprognosen im Allgemeinen sind für die Energiewirtschaft und die Liberalisierung des Elektrizitätsmarkts von besonderer Bedeutung. Um einen effizienten, sicheren und wirtschaftlich optimalen Betrieb in einem Energiesystem zu erreichen, ist es notwendig, das Energiesystem und den dazugehörigen Strombedarf rechtzeitig zu analysieren und schnell auf Änderungen der elektrischen Last zu reagieren. Diese sind in hohem Maße von einer genauen kurzfristigen Lastprognose abhängig.

Auf der anderen Seite ist eine langfristige Lastvorhersage auch für die Energieversorger und andere an der Stromerzeugung, -übertragung, -verteilung und -marktbeteiligten bei der Planung und Strukturierung von großer Bedeutung. Eine zeitnahe Implementierung langfristiger Lastprognosen hilft dem System, die Netzwerkstabilität zu verbessern, die Ausfälle von Geräten und Stromausfällen zu reduzieren und eine zuverlässige Energieversorgung zu gewährleisten.

Diese Dissertation versucht, einen neuen Ansatz für das Problem der Lastprognose vorzuschlagen, um genauere Ergebnisse mit einer geringeren Fehlerquote zu erreichen. Dazu wird eine Methodik eingeführt, die auf den Theorien *Chaos* und *Concept Drift* und Zeitreihenanalyse basiert und der *Adaptiven Neuro-Fuzzy-Inferenzsystem (ANFIS)* die Hauptrolle beim Trainieren und Testen des Modells spielt.

Als Fallstudie wurde die elektrische Last des Clausthaler Umwelttechnik Instituts (CUTEC) aus dem Jahr 2014 ausgewählt. Diese Daten wurden basierend auf der *Mutual Information Funktion*, die von Concept Drift Theorie inspiriert wurde, analysiert, um die Ähnlichkeiten und Wiederholungen darin zu finden. Diese wurden danach dazu dient, die Daten zu kategorisieren. Danach wurde die Existenz von Chaos in der Zeitreihe mithilfe von drei Verfahren, nämlich *Embedding dimension, Grassberger-Procaccia correlation dimension* und *Lyapunov exponents* untersucht. Eine chaotische Zeitreihe basiert auf einem deterministischen System, d.h. es enthält genügend Informationen, um durch die in dieser Dissertation vorgeschlagene Methode vorhergesagt zu werden.

Der Training- und Testprozess wurde mithilfe von ANFIS in MATLAB und die Vorhersage für drei verschiedene Zeithorizonte durchgeführt: Wöchentlich, monatlich und saisonal. Die Prozentsätze des absoluten Fehlers des Erwartungswertes (MAPE) als einer der wichtigsten Indizes in der Lastprognose liegt in allen Zeithorizonten zwischen 2,1% und 2,6%, was deutlich niedriger als die meisten berichteten ähnlichen Fällen in anderen Studien sind und das vorgeschlagene Verfahren zeigt einen effizienteren, zeitsparenderen und präziseren Ansatz für die Vorhersage der elektrischen Last.

Darüber hinaus beinhaltet diese Forschung ein Simulationsmodell für ein dezentrales Energiesystem (DES), das in MATLAB und SIMULILNK implementiert wurde. Der Benutzer kann die elektrische sowie thermische Last als Zeitreihe über eine Benutzeroberfläche hinzufügen und die Priorität und zusätzliche Details der Anlagen definieren, wie z. B. die Nennleistung, den Wirkungsgrad oder die Speicherkapazität. Das Model liefert Informationen über den Fahrplan jedes Moduls, das auch als Excel-Datei exportiert werden kann. Zudem werden auch verschiedene grafische Darstellungen wie z.B. Speicherstatus generiert. Dieses Modell wurde mit dem Energiepark im CUTEC-Institut validiert. Die Simulationsergebnisse für das Jahr 2014 wurden dann mit den Ergebnissen einer kommerziellen Energiemanagementsoftware (EnergyPro) verglichen. Die Ergebnisse lagen in der gleichen Größenordnung mit einem vernachlässigbaren Unterschied von weniger als 1%.

Contents

List of Figures

List of Tables

Abbreviation

AF	–	Adaptive Filtering
AI	–	Artificial Intelligence
ANFIS	–	Artificial Neuro Fuzzy Inference System
ANN	–	Artificial Neural Network
CHP	–	Combined Heat and Power
CUTEC	–	Clausthaler Umwelttechnik Forschungszentrum
EMS	–	Energy Management System
FIS	–	Fuzzy Inference System
GA	–	Genetic Algorithm
GDP	–	Gross Domestic Product
GEF	–	Global Environment Facility
GUI	–	Graphical User Interface
LSE	–	Least Square Error
MA	–	Moving Average
MAPE	–	Mean Absolute Percentage Error
MATLAB	–	MATrix Labratory
ME	–	Mean Error
MENA	–	Middle East and North Africa
MLP	–	Multi-Layer Perceptron
MLR	–	Multiple Linear Regression
MSE	–	Mean Squared Error
PCM	–	Phase Change Material
PV	–	Photovoltaics
RLSE	–	Recursive Least Square Error
RMSE	–	Root Mean Square Error
STDE	–	Standard Deviation Error
SVM	–	Support Vector Machines
SVR	–	Support Vector Regression

I. Chapter 1

1. Introduction

The concerns about the increased exploitation of the natural resources and shortage of them in future and also the dilapidation and pollution of environment due to the expansion, construction and operation of energy systems have generated a growing interest in the improvement of energy efficiency. In other words, the new energy systems should be designed in a way by which the consumers' needs are fully satisfied while natural resources exploitation and environmental pollutions are minimized through efficient implementation of renewable energy systems. Therefore, different optimization methods and procedures have to be tested and applied on the systems. To perform such tests, a profound understanding of an energy system mechanism and its modelling procedure is indeed very important and necessary. The simulation runs of the model provides us with valuable information such as system responses to different changes and the appraisal of optimization methods, prior to real-life application of such methods to real systems (Rothe, 2010).

On the other hand, the supply of electricity in sufficient quantity with a favorable quality at competitive price together with a reliable supply through a stable network is of vital importance for the development of industries, regions and countries. Precise electricity consumption forecast plays a vital role in the modern energy planning and has primary importance in energy system modellings and simulations. The design, expansion plans and construction of electricity transmission network and distribution facilities entail an investment of billions of dollars, which makes the electrical consumption forecasting a very important factor for the investors and companies (Rothe, 2010).

The load prediction is also of great importance for energy suppliers, independent system operators, financial institutes and other parties involved in power generation, transmission, distribution and market. For the energy suppliers, a timely implementation of load forecasting helps the system to improve the network stability and reduce the equipment failures and power outages and ensures a reliable energy supply (Li, 2014). It can also help them by the flexibility studies of their power plants.

Furthermore, the load forecasting is a fundamental and inseparable process in the planning and operation of electrical facilities. The utility companies use the forecasted load curves for controlling processes and decision makings like optimized energy production and distribution planning, designing network expansions, maintenance and service planning, fuel allocation and offline network analysis (Kothari, 2008).

An accurate forecasting of power system load is also a very important issue in the economic studies and performance management of a power system. In general, the load forecasting is performed based on the consumers' behavior and the development of power system networks. The companies need to estimate the future load pattern based on the various possible scenarios for load management and economic decision making processes, such as capital investments, risk management, operating costs reduction, assessment of profit margins, revenue losses, decisions on energy sales and customer satisfaction. In the market, the accurate prediction is the basic of electric energy trading and setting of the electricity price. Therefore, a correct prior knowledge of the future load characteristics is necessary for the marketing (Kothari, 2008).

With the growth of power system networks in recent years and increase in their complexity, various parameters have become influential in the electric power production and also the "demand and load management". Likewise, there are many different factors that influence the load pattern among which, the following parameters are particularly worthy of note: Economic factors, time, weather, energy price, sudden disturbances and other unpredictable factors (Gómez-Expósito, 2016).

As a matter of fact, the load forecasting has various important applications and therefore, the estimation of the load demand must be as precise as possible. This has been a great challenge for many decades. In this regard, a change in the load forecasting techniques from the classic methods toward the intelligent systems has been featured in many different studies and many new methods have been introduced in the recent years. The evaluation of these methods can be done based on the accuracy of their results.

The predictive models are widely used as examples of such intelligent methods. These models need series of data as input, to predict the future load in the intended system based on them. The selection of these data can be done experimentally or based on the intelligent systems. One of the most popular and recent approaches in predictive methods are the Artificial Neural Networks. The classic methods, basically, rely on statistics. However, due to nonlinear behavior in load distribution networks, using intelligent methods is a better choice for the load prediction. Using neural networks was the beginning of a new era in load forecasting, which improved the accuracy of prediction to a great extent, but still the modern energy system management demands more accurate approaches, which has resulted in huge efforts in this field in the recent years. This dissertation aims to introduce a new approach to load forecasting by combining several known theories in order to achieve a more precise prediction in comparison to current methods.

2. Motivation

According to the German Association of Energy and Water Industries (BDEW), the electricity produced from renewable energy sources has grown in 2016 again (190 Mrd. Kilowatt hours) and was about 30 per cent of gross electricity generation in Germany. Due to the rapid development of energy systems based on renewable resources and their facilities and equipment, such as photovoltaics and wind turbines, new challenges have arisen, e.g. the fluctuating electricity production or the decrease in electricity consumption at the times with the maximum electricity generation capacity. There have been many studies and projects in the last decades to find a solution for challenges. Introduction of new methods for energy storage including new ideas for heat storage or power to X projects were some of the most important developments in this field. However, to apply such methods, having a model of energy system and performing the optimization procedure on such model is necessary.

During the last few years a wide range of structural changes have been done in the economic planning and operation of power systems around the world, which are usually known as “reconstruction” or “deregulation” of the power system. With the advent of deregulated energy systems, the conventional concepts of power systems operation has been altered and new definitions have been emerged. On the other hand, the concepts of consumer participation and load response in the network is undergoing enormous changes. Using the traditional structure in power systems reduces the efficiency, because the electrical network monopoly eliminates the motivation for an efficient performance and makes unnecessary and inefficient investments. The costs of such mistakes will be eventually passed on to the consumers.

To solve such a problem, various solutions have been proposed. In the past, the researches focused more on the formulation of some practical limitations, such as voltage range, generation limits, transmission lines capacity and other potential constraints. But the new restructuring in the power systems has required the separation of three main components of the electric power industry: generation, transmission, and distribution (Hooshmand, 2013).

The separation of transmission ownership from the controlling process is one of the best ways to make a competitive market for the power generation. An independent operational control, on the other hand, cannot be guaranteed without an independent system operator. The system operator needs to be independent from all of the market participant, such as energy suppliers, transmission owners, utility companies and end users.

The system operators establish rules on energy and service markets and try to operate system in a fair and non-discriminatory way to achieve an efficient and yet competitive market in a high quality and reliable network. In order to operate the system in this way, the operators must be equipped with powerful computational tools and also very accurate information about the future load. On the other hand, the supplying companies have to provide energy to the consumers with high reliability and high quality. At the same time, they have to consider other issues, such as environmental protection and optimized usage of primary energies together with optimal utilization of the equipment in power plants, transmission and distribution networks (Rothe, 2010) (Hooshmand, 2013).

An accurate load prediction can help the operators and suppliers to raise the efficiency and quality of their operation. The electrical engineers have long been familiar with this concept and the reconstructions in electricity industry has introduced new concepts in this field.

The accurate prediction of the electrical load has always been of great interest for the researches. The prediction may be short-term which is more detailed and can be used for operational purposes. For such purposes accuracy is very important. For instance, a short-term load prediction has to be performed to adapt the electricity generation to the electricity consumption. On the other hand, the long-term load forecasting is the basis for the energy investments and other long-term planning, such as the principal considerations of power system development or extra-high voltage systems expansion which involves irrevocable investment decisions (Ding, 2016).

There are many different physical factors which influence the load characteristics. These factors are non-linear, very complex and some of them have more influence in comparison with the other ones. To achieve a more precise and better load prediction, such factors should be taken into consideration (Hooshmand, 2013). The electricity demand pattern is mainly affected by several factors which can be categorized into four main groups:

- Time
- Environmental conditions
- Social and economic factors
- Random disturbances

Being influenced by one or more of these factors, the pattern will form complex variations (Rothe, 2010) (Islam S. M.-A., 2001). Most of the modern methods in load forecasting are based on the Artificial Neural Networks (ANNs), since they have a very effective and accurate self-learning characteristics. However, these methods normally consider only a part of these influencing factors as their input and try to train their model and predict the future based on them which is exactly their drawbacks. Since the training of the model is only based on a part of those influencing factors such as weather conditions, only some of the important parameters are highlighted and the rest will not be considered. This leads to neglecting a huge part of information and making predictions based on limited information. For instance, there are many studies, which consider the weather temperature and humidity as two main sources as training parameters and the future prediction. On one hand, if only the historical data of these parameters is used for the prediction, there is no guarantee that the same weather condition will occur again in future. On the other hand, if the predicted data of these parameters is used for training the ANN system, there is the risk that those forecastings are of course susceptible to larger errors, since the nature of weather conditions is random. (Danladi, 2016) (Park, 1999).

In contrary to the above mentioned techniques, this research tries to introduce a new method based on data mining on the electrical load data itself. Since all of the parameters have already made their influence on the load pattern, the time series already include all the information needed to predict the future load. In other words, to predict the future pattern, it is only necessary to detect the concepts and information in the time series, classify the input based on that and prepare the data for a systematic input in the training process. This classification and systematic process together with the Artificial Neuro-Fuzzy Inference System (ANFIS) as the training method make the proposed approach in this dissertation different from the previous and on-going works in the similar field, which provide very accurate results with almost less than 2 percent Mean Absolute Percentage Error (MAPE) in short, mid and long-term load forecasting. Since this method only takes the past electrical load data as input, it can also be very helpful for the systems with no or limited information available on other historical data influencing the electrical load of the system such as different weather conditions.

3.General Philosophy of Prediction

During the history of science, the scientist have always tried to foresee the future incidents precisely to achieve true information and prove their ideas in many different fields. Needless to say, in practice, none of them were successful and they all came to the conclusion, that our information even about the present is not enough. As a matter of fact, although there is a wonderful order in the nature, but our knowledge about all the aspects of this order is very little. Thus, in the lack of empirical and experiential data, the mankind has tried to recognize and model some of the behaviors and processes in nature and universe to be able to predict the future, but in reality these processes cannot be modelled easily due to complexity. The human behavior itself and its relation to groups and other societies is an example of such complex systems. There have been some models, which tried to understand and simulate these behavior and relations with chaotic or turbulent models, but eventually those models can just be helpful for less intense changes or smaller groups and communities. Undoubtedly, it is almost impossible to model all the processes in the chaotic world around us; however, applying some simplifications may help to create models, which may be very useful to understanding these processes and eventually finding new methods to have an approximate prediction of the future (Pillkahn, 2008).

There are always some constant elements in each process which are known in advance, but it is almost impossible to collect enough of these elements to achieve a definite and exact prediction of the future. The lack of such elements, cause uncertainties in the process, which makes the prediction a challenging task in most cases.

Prediction in different fields can have different concepts and meanings. For example, if asked a theorist, who believes in total randomness in the world system, about the future prediction, he would say that it is an impossible thing, because his task is to analyze different processes and build a feasible model out of different elements with multiple degrees of freedom, high complexity and dependence. In an astronomer's point of view, it is possible to predict the future, because he can calculate exactly how many solar eclipse will happen in the next hundred years with an accuracy of tenth of a second. An Economist would say that the future prediction is possible to some extent. He can estimate different economic indicators, such as gross domestic product and based on that he can predict the economic growth in future with a relative good accuracy based on the estimated data. A meteorologist would probably say that he can properly predict tomorrow's weather, however there might be times that his prediction is not accurate (Wack, 1985).

There are two decisive elements in a prediction process: The change spectrum and the knowledge spectrum.

The change spectrum has four different states:

- *Constant state*: In this state the variables, which define the behavior of the system, are in stable conditions, e.g. physical constants.
- *Focused change*: This type of changing is normally related to the area of trends which is predictable, e.g. cyclic changes, evolutionary changes.
- *Unfocused change*: This is the area of uncertainties. This type of changes are exemplified by economic, financial systems and scientific discourse without clear answers.
- *Chaos and turbulent change*: This states refers to a turbulent and chaotic systems, in which the changes cannot be foreseen exactly. (Pillkahn, 2008)

The knowledge spectrum has also four different states:

- *Knowledge*: This state is based on our knowledge of different science and the proved facts related to them, e.g. the laws of thermodynamics.
- *Theory*: In this state, there is sufficient evidence to establish a hypothesis and while an accurate and exact proof is not possible, the evidences support the hypothesis.
- *Supposition:* It is possible to reach a conclusion based on the evidences and assumptions in this state, and clarify the uncertainties based on a chain of facts.
- *Speculation:* This state is typified by hopes, wishes, believes and fears. Here some conclusions can be proposed based on speculations and predictions. (Fayyad, 1996)

Figure I-1 shows the change spectrum which can vary from a constant state to completely unknown change. As a result, we may confront uncertainty when we try to predict the future. Our understanding of future depends on the state of knowledge about the changes in future, which can vary from facts to speculations. For instance, predicting the eclipses is within the range of our knowledge, therefore it is possible to determine the exact time of an eclipse with a very high accuracy. The economic growth is a kind of change which can be predicted by proposing different probabilities. The weather condition, however, belongs to another group of changes, and the weather forecast always includes supposition and guess. The movement of a double pendulum is among the changes which is impossible to predict, although an almost specific range of motion can be predicted. The condition of total randomness and uncertainty is typified by Tsunami, where it is not possible to determine the time, location, and even the intensity. (Fayyad, 1996)

The Load forecasting cannot be allocated to just one category, because it is highly dependent on the time horizon of prediction (related to change spectrum) and also the available information (related to knowledge spectrum) about the system being analyzed.

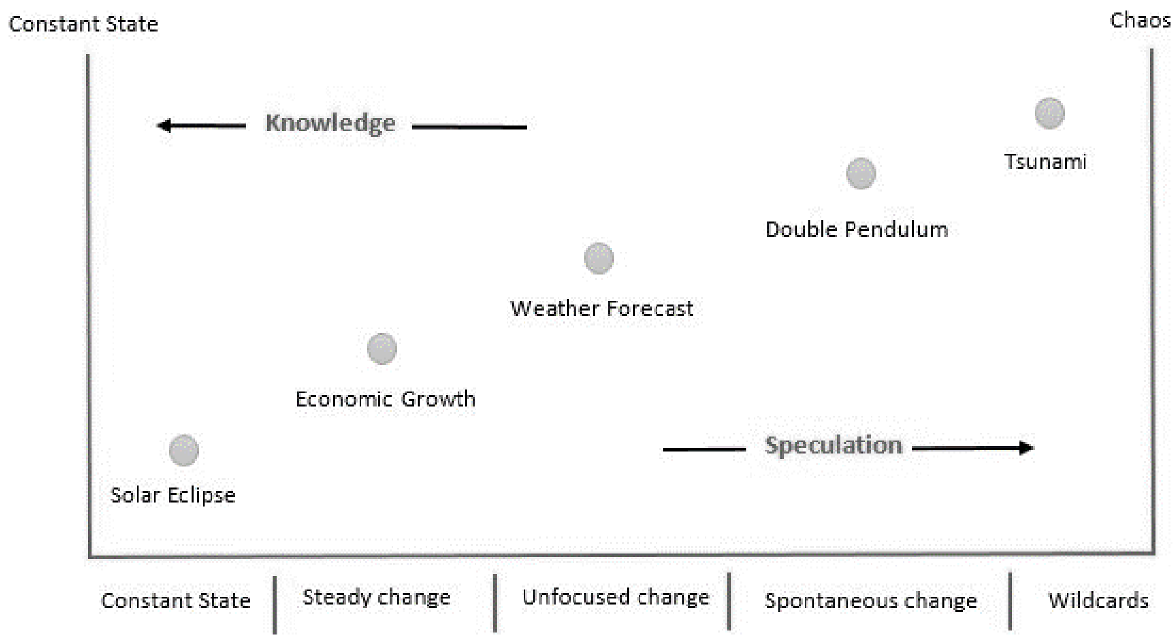


Figure I-1: The spectrum of change from constant state to chaos state (Pillkahn, 2008)

In general, the present is the result of the incidents, which has happened in the past and the future will be created as a reaction to the incidents happening in the present based on the "causal logic" or "cause and effect principle"[1]. These reactions are time dependent, i.e. an action may cause different reactions under certain circumstances. The future will be influenced by such reactions and other unpredicted factors. Figure I-2 shows these relations:

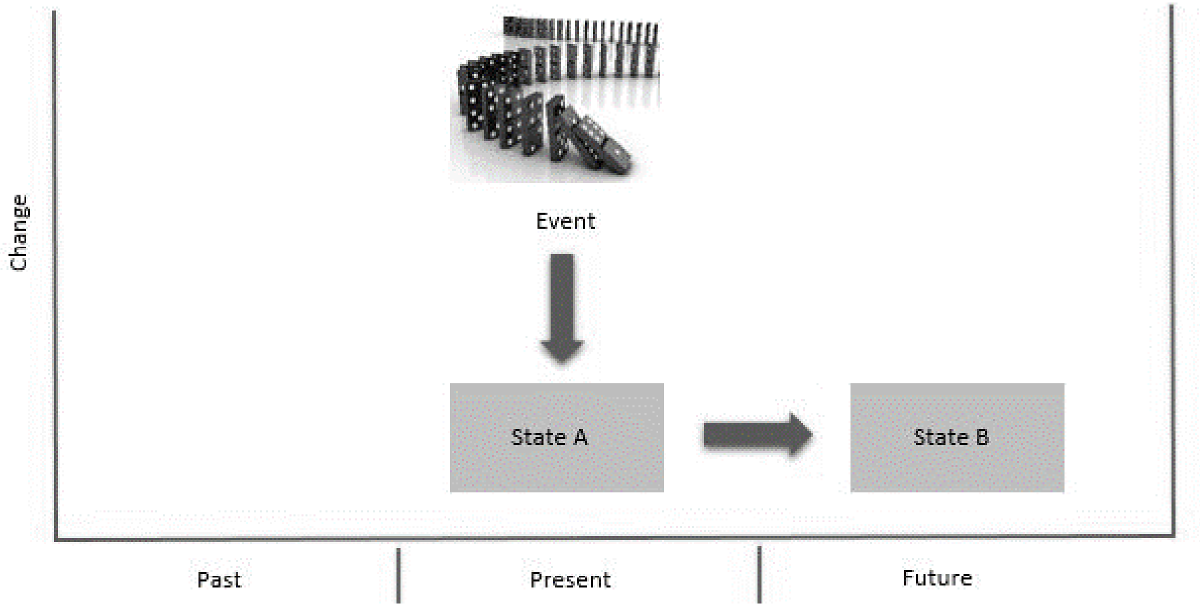


Figure I-2: Causal relations: If incident X happens, then the state A will change to the state B! (Pillkahn,

The model presented in this figure is very simplified. Such a model may be useful for the short-term changes or very simple causal relations. For more complicated systems with more complex relations, an improved model is needed to achieve better data processing. In such models, the second and third level of effects will also be considered. (Figure I-3)

[1] Causality (also referred to as causation or cause and effect) is what connects one process (the cause) with another process or state (the effect), where the first is partly responsible for the second, and the second is partly dependent on the first. In general, a process has many causes which are said to be causal factors for it, and all lie in its past. An effect can in turn be a cause of, or causal factor for, many other effects, which all lie in its future. Causality is metaphysically prior to notions of time and space. [Wikipedia]

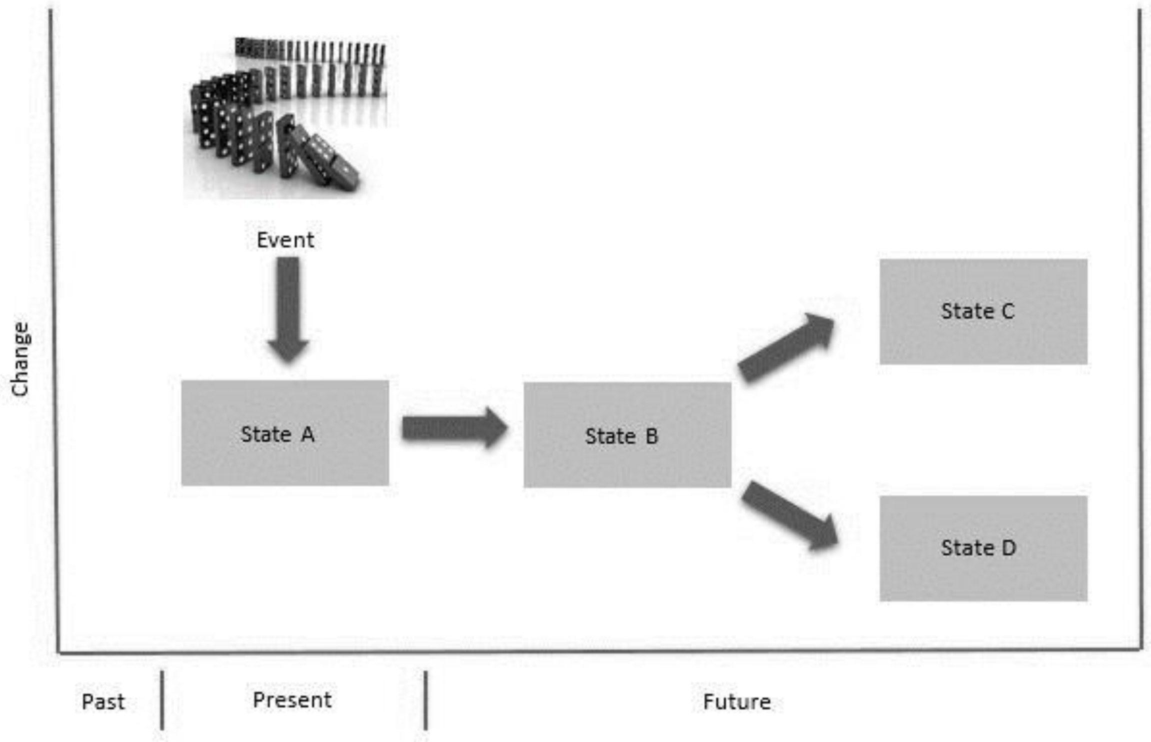


Figure I-3: Causal relations for the long-term changes: The second and third level effects on the future incidents (Pillkahn, 2008)

There are two ways of getting information on the upcoming incidents. Firstly, it is possible to predict the future based on the repeatability of the observed events in the past and secondly, to forecast the incidents in future considering the most influencing factors. The rules and laws are good examples of the repeatability. It means that, knowing that the state ‘A’ turns to state ‘B’ based on the law ‘X’, state ‘AA’ turns to state ‘BB’ based on the same logic which is unquestionably under certain circumstances. Figure I-4 shows the role of a law and its application in the prediction of future changes:

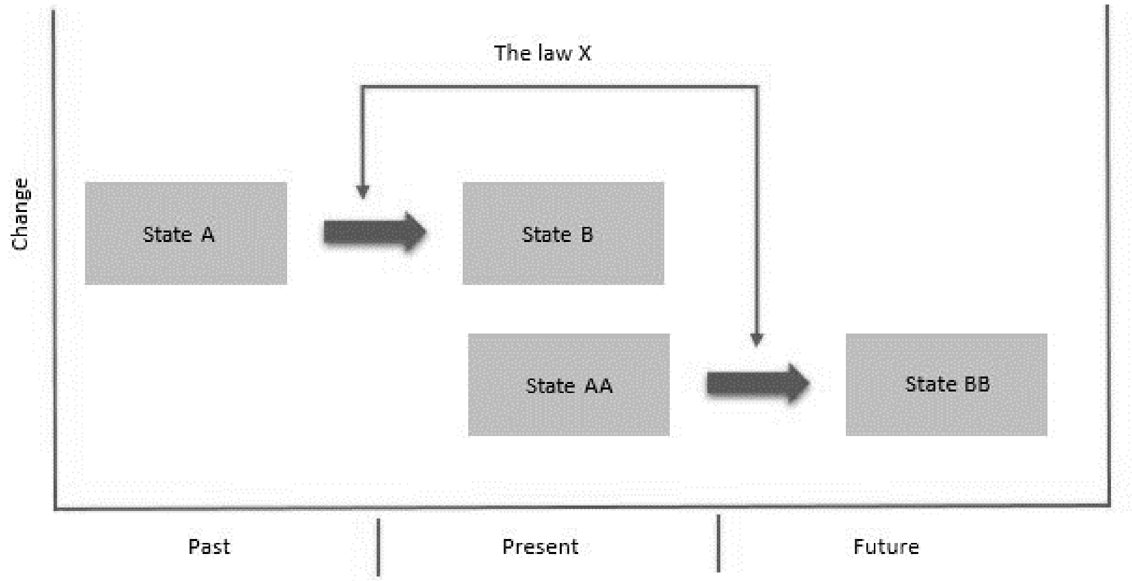


Figure I-4: Knowing that the state A turns to state B based on the law X, state AA turns to state BB based on the same logic (Pillkahn, 2008)

One of the ways to improve the basic prediction model is to apply the techniques based on the time series data processing to include the second type of information gathering methods mentioned above. The methods associated with the time series are very extensive and widespread. In the prediction science, they are called trend analysis.

The term “trend” refers to *something* that tries to describe and recognize a certain direction in which the system seems to be going. Here, the “something” is the measurable elements of a certain development in the system.

The trend analysis can be quantitative based on the statistical tools. But as we know, the accuracy of the prediction will decrease drastically as the considered time interval increases. Therefore a qualitative analysis is more accurate for longer term considerations. Such analysis includes supporting explanations. In trend analysis, certain characteristics of the change in system will be recognized based on the past incidents and events. These characteristics will then be passed on and applied to determine the same change in future, for example to calculate the slope of the change curve by inference and extrapolation (Figure I-5).

The time series can also be helpful for the medium-term analysis as long as they assume a constant development or a steady progress for the influencing factors on the predictable future trends.

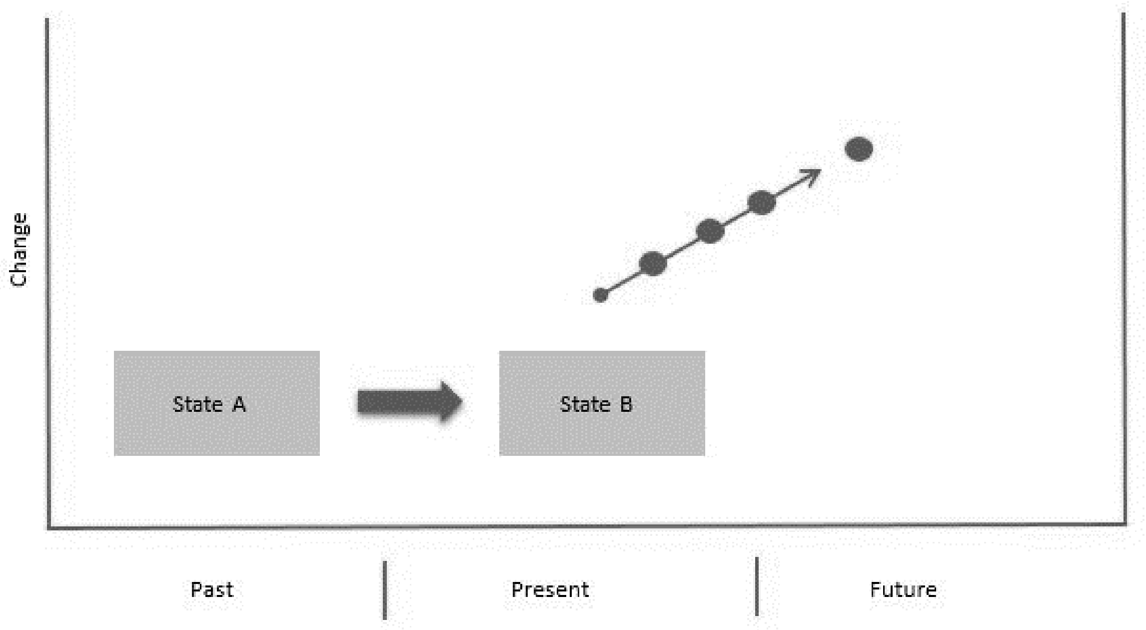


Figure I-5: Time series: The changes in the past, can show us the future!

It is worth to mention that the prediction in general needs different basic principles and methods in regard to the time interval it tries to forecast. Table I-4 shows an overview on these basic principles in the analysis.

Table I-1: Basic principles in the future analysis

Basic Principles	**Orientation**	**Time Interval**
Causal Logic	Focusing on the past	Short to Mid-Term
Time Series	Focusing on the past	Short to Mid-Term
Theories and Laws	Valid for ever	Mid to Long-term

Indeed, combining these basic principles and developing new methods based on them, helps to extract more precise information about the future and finally to develop a better and clearer image of the future. Based on this table, a combination of causal logic based on the laws in time series, is the best way to achieve the most accurate results (Alfares, 2002) (Hong W.-C. , 2011) (Tascikaraoglu, 2011).

This research takes the time series analysis method as the basic and the casual logic as the supporting principles toward the problem of electrical load forecasting for different time horizons.

4. The Research Structure

This dissertation focuses on the forecasting of the electrical load data with more accuracy and a lesser error rate compared to the current methods. For this, a methodology based on the chaos and concept drift theories will be introduced, in which the Artificial Neuro Fuzzy Inference System (ANFIS) plays the main role in training and testing the model.

The following Chapter introduces the chaos theory and nonlinear systems, with emphasis on their application in time series identification. Proving the existence of chaos in the sample data series is the first step in the algorithm proposed in this dissertation. Therefore, the chaotic nature of the considered sample time series will be examined by three different methods. This chapter also describes a kind of anomaly existence in time series, which is the concept drift. With the help of this phenomena and some supporting functions, the sample data will be classified based on the detected similarities. In this way the datasets will be ready to be fed to the training and processing part systematically. The third chapter will briefly provide the basic information on electricity load prediction in general, introducing the classic and modern approaches toward the problem of load forecasting. Afterwards, the most common modern method, namely Artificial Neural Network (ANN) will be studied and the result of prediction based on a simple ANN will be presented. The fourth chapter describes a tool for analyzing the use of contextual information by fuzzy neural network. Here, the ANFIS model will be introduced and the categorized datasets based on concept drift will be fed to the ANFIS for the training and testing processes. Finally, the results of electrical load prediction based on this model will be presented. These results will be compared to the ones from the previous chapter (simple ANN-based method) to show the privilege of the proposed model in this dissertation over the other methods.

The next chapter is dedicated to introducing a simulation model for a decentralized energy system. Here, the importance of such modelling will be discussed and the case study of the Energy Park at CUTEC institute will be introduced. Different units and basic elements of this simulation model including CHP units, boiler, photovoltaics, electrical and heat storage units will then be reviewed and the sample electrical load of the year 2014 will be fed to the simulation model and the test results (year 2014 and sample cold and hot week/day) will be presented. The same system will then be modelled in the EnergyPro software. The results of these two simulations will then be compared to validate the implemented model in MATLAB.

Subsequently, a chapter will introduce the current energy system in Iran. Following by the future development plans in renewable energies in Iran, the transferability of the proposed model to Iran's network system will be discussed. The Conclusion, final remarks and directions for future work are given in the final Chapter.

II. Chapter 2

This chapter starts with an introduction to the concept drift theory in time series and its definition in the electrical load data. Following that, the chaos theory in the non-linear systems will be presented and some methods for the detection of chaotic systems will be introduced. These methods will then be applied on the empirical electrical load data of CUTEC Institute in the year 2014 which is the case study of this dissertation. Proving the existence of chaos in the load data, the lags needed for the classification of them will be calculated and the input data will be prepared to be inserted to the ANFIS-based training and testing model.

The main architecture and components of the load prediction model proposed in this dissertation is shown in the figure II-1. This system is designed to train, test, regulate and evaluate the fuzzy neural network with the ability of anomaly detection.

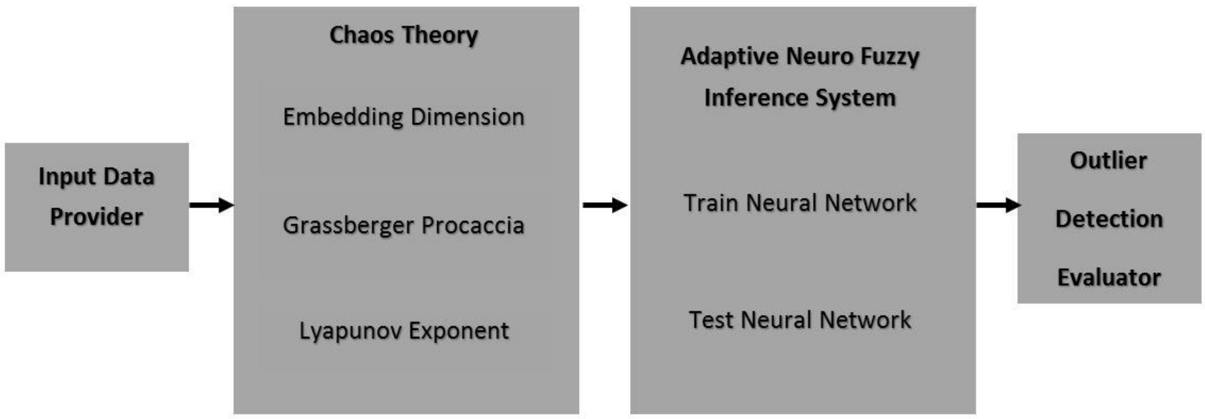


Figure II-1: The main architecture of the prediction system based on the fuzzy-neural network (own representation)

Based on this system, the main structure of the implementation can be divided into the following main steps:

1. Input data provider which is the classification of input data based on the concept drift. This part will be discussed in this chapter.
2. Proving the existence of Chaos in the selected electrical load data which will also be covered in this chapter.

3. Implementation of the load forecasting model based on the fuzzy neural networks using the prepared input data from the first and second step. This part will be discussed in chapter 4.
4. Comparing the proposed method in this dissertation with the previous methods, such as simple neural networks. (also in chapter 4)

The mentioned steps show eventually the importance and effectiveness of the artificial intelligence methods and specially the Adaptive Neuro-Fuzzy Inference System (ANFIS) in the electrical load forecasting systems.

1.Concept Drift

The digital world is expanding rapidly and to deal with the huge amount of the new incoming data, very effective methods should be developed for the analysis of the systems receiving them. These data normally come in the form of streams. Saving all of the incoming stream data is inefficient and in some cases infeasible. To process all the incoming data, the predictive models need to be trained continuously or by being retrained using the recent sets of data.

The first step in analyzing the system is recognizing the main concept in the data. When the system is time invariant, this task seems to be very simple. Yet, one important challenge in stream analyzing arises, when the source of the data is not stationary and the data distribution change over time. In this case, the concept to be learned for analyzing the system is also a function of time. This phenomenon is called the “concept drift”. Concept drift refers to the unforeseen changes in the output variables, while the input characteristics may stay unchanged. A typical example of concept drift is the behavior of the customers in online shopping. Since a daily or weekly sales prediction based on the amount of money spent or advertising promotions could be possible and effective, the model is likely to become less accurate over time by changing season. (Chi Xie, 2006) Another typical example of concept drift is the change in workload profile of a system, which controls the load redistribution in computer clusters. (Widmer G., 1996)

In power engineering the electrical load can be recorded in form of stream data and therefore, it is possible to analyze this data based on adaptive learning methods. In electricity load measurement the concept drifts are caused by the changes in consumers’ behavior during holidays, social events or different weather conditions which can be divided into two main types. The first type is permanent or temporary changes, which may be caused by the change of economical factors. The second type is caused by seasonal changes such as changes in weather condition or the amount of daylight. These two types of concept drifts should be considered in the load forecasting model. (Gama, 2013) (Chicco, Napoli, & Piglione, 2001)

One of the important issues in the electrical load stream is the concept of repetition which can also occur in form of concept drifts as parts of the data stream. With the help of some specific rules and relations in studying the probability of such repetition, it is possible to estimate the occurrence and recurrence of the concept drifts. In this case by applying the adaptive learning, the model can update itself faster by learning the concept drifts and eventually the error rate will be reduced significantly. (Maayan Harel, 2014) (João Gama, 2014)

Figure II-2 shows the electricity load data where some repetitions can be specified easily. These repetitions could be signs of concept drift in time series, based on which, a very simple classification of data is possible.

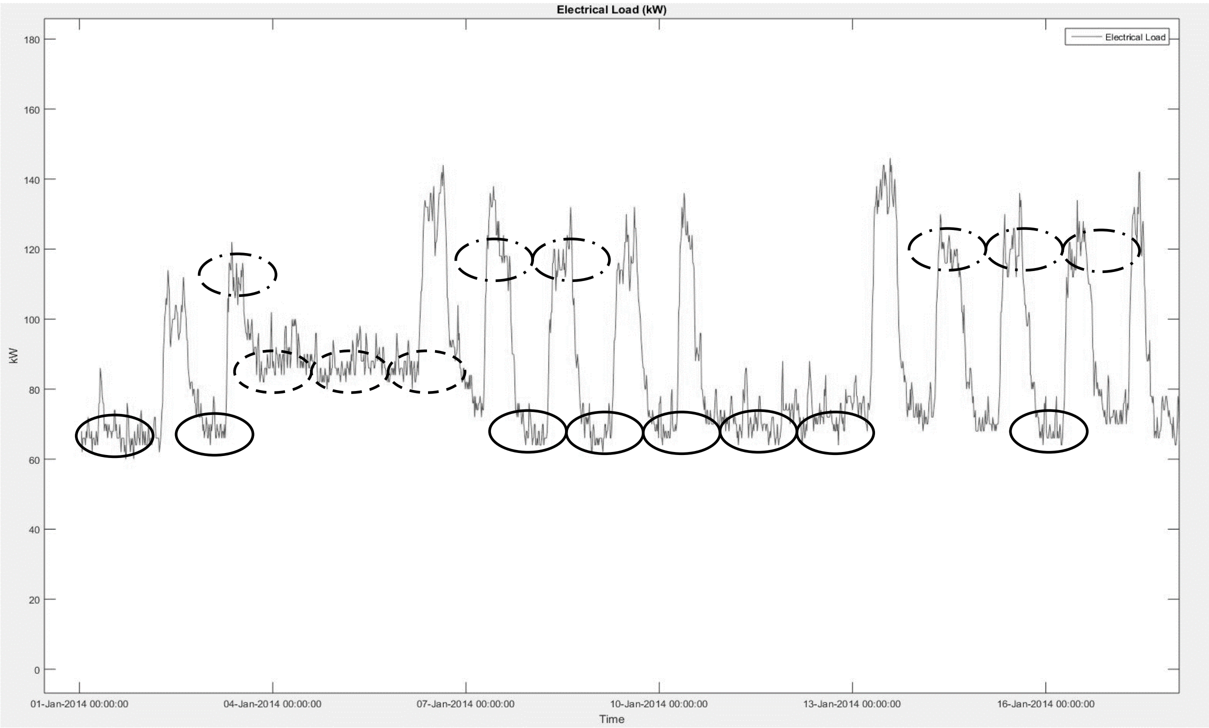


Figure II-2: An Example of the concept drift and repetition in a sample electrical load data (own representation)

This study intends to perform the load forecast based on the repetition in the possible concept drifts in the load data stream. For this purpose, initially the sample time series will be analyzed by the *mutual information function* which will be discussed in part 3.1. in details. In this way, the input data will be classified based on the triggers on the concept drift or the so-called “lags”. These lags create datasets from the main time series and prepare the input data for the training process.

One of the important basics of this work is the theory of *Chaos*. It is possible to identify the systems, which are based on deterministic chaos with this theory and the related mathematical tools. Being deterministic means that the system is predictable and it is not based on some random numbers. In the next part, the chaos theory will be introduced, and the existence of deterministic chaos in the sample electricity load of CUTEC in 2014 will be examined by different methods.

2. Chaos

For many years, the uncertainty has been modelled by applying random variables and processes in probabilistic models, but perhaps one of the most important facts derived from years of researches and studies on non-linear dynamical systems, is that uncertainty can also be caused by purely deterministic mechanisms.

Even simple deterministic non-linear systems can cause behavior which may be hardly distinguishable from a complete random process. Considering this, there has been many efforts to describe complex system by simple non-linear models. This has given rise to the “chaos phenomenon” and variety of related techniques for studying such chaotic systems.

Chaos is the science of nonlinear and unpredictable systems. In contrast to the predictable phenomena like electricity, gravity or most parts of the physics, the chaos theory deals with the complicated nonlinear systems in which, a slight amount of uncertainty about the initial state can be amplified to a total lack of knowledge about the future states. Such systems are almost unpredictable or at least impossible to control. The turbulence, weather and market are some examples of such systems.

The mathematical phenomenon of chaos is studied in many different fields such as population biology, economics, astronomy and psychology. The chaos theory is focused on the phenomenological behavior of dynamical systems which are highly sensitive to the tiniest changes in initial conditions and show random and unpredictable behavior. Having said that, according to this theory, in spite of the apparent randomness in the behavior of such systems, there are underlying patterns, repetition, constant feedback loops, self-similarity, fractals and self-organization, which can help to understand the behavior of such systems (Bishop & Zalta, 2017) (Wu Zhi-jun, 2013)

2.1. Chaos Phenomena in Time Series

One of the most important application of the chaos theory is within the field of time series analysis, in which the non-linear dynamics offer a new class of models. Such system modelling based on the stochastic paradigm may lead to forecasting algorithms with a significantly higher accuracy in comparison to traditional linear stochastic models. (Kellert, 1993)

Time series generated by chaotic systems

Chaos is a nonlinear behavior in a range between oscillatory and random behaviors. It has been proven that a mathematical chaotic system has a deterministic origin, i.e. the future behavior of a chaotic system can be fully determined by the initial conditions, with no random elements involved. (Stark, 1994) Although this deterministic nature does not make such systems fully predictable, it is possible to define the state of a chaotic system at any moment by:

$$X(t)= (x(t), x(t - \tau), x(t - 2\tau), \dots , x(t-(k-1)\tau) \tag{2.1}$$

Where τ is the sampling interval and $x(t)$ describes the state of the deterministic dynamical system at a given time t. The system states will be specified by a map $x(t_0) \rightarrow x(t)$, given that the initial state was $x(t_0)$ at t_0. However, in most cases this map is unknown and it is only possible to observe some measuring function F. This function can be the measurement of any variable in system, such as temperature or electrical load which can only be done in discrete time intervals (τ)[2]. Assuming that F is a non-linear function of system, for each $(t + \tau)$, it can be written:

$$\begin{gathered} F= \mathbb{R}^k \rightarrow \mathbb{R}^k \\ X(t+\tau)= F(X(t)+p(t)) \end{gathered} \tag{2.2}$$

where $p(t)$ is the probability function with an average value of zero, considered due to the possible errors, such as rounding and measurement inaccuracies. Based on the mathematical facts, in deterministic systems, the difference between two adjacent states (error) remains either

[2] In some cases it is possible to measure more than one variable of the system. On obvious consequence of this, the system will be described through multi-variable time series.

very low (implying normal and stable systems) or it grows exponentially (implying chaotic systems), while in random systems, this difference (error) is distributed fully coincidentally.

A simple example of a chaotic system is the Henon map, which takes a point (x_n , y_n) and maps it to a new point under the following formula:

$$x_{n+1} = 1 - a\, x_n^2 + y_n \qquad (3.a)$$
$$y_n = b\, x_n \qquad (3.b) \qquad (2.3)$$

The Henon map depends on *a* and *b* as the main controlling parameters. In *classical Henon map* the values of *a* and *b* are taken to be a = 1.4 and b = 0.3, so that the system is chaotic. For other values of *a* and *b* the system may be chaotic, intermittent or a periodic orbit.

Figure II-3 shows the observation of the x coordinate in this time series with the observation function $F(x_n , y_n) = x_n$.

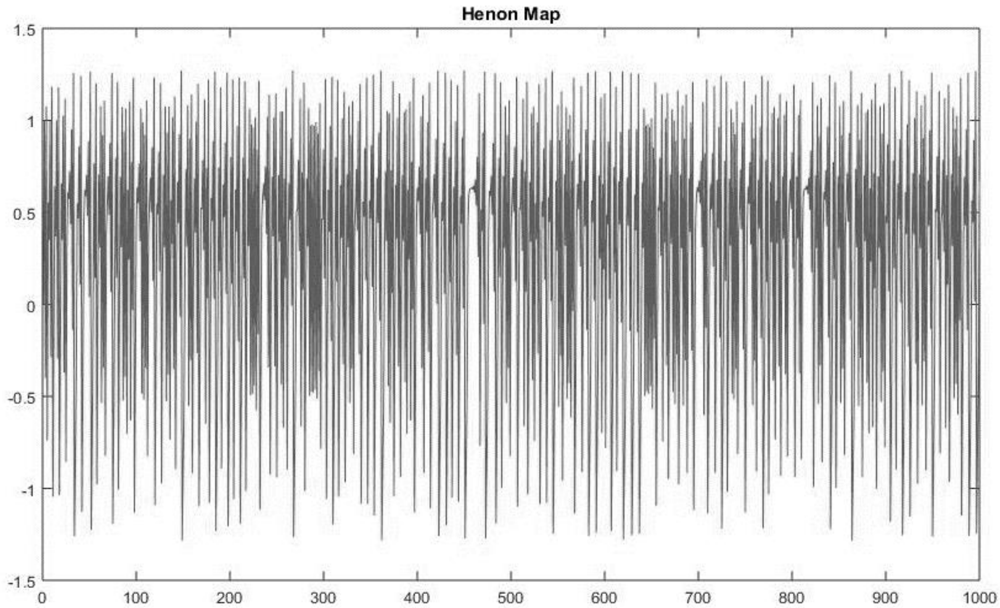


Figure II-3: x coordinate of Henon map – a chaotic system

Another example of a chaotic time series is the Lorenz equations, which were introduced by Edward Lorenz in 1963 to develop a simplified mathematical model for atmospheric convection. These equations are still accurate and widely used. Lorenz was the first person, who discovered that it is possible to generate Chaos with computers. Figure II-4 shows the z co-ordinate observation of, which are defined as below:

$$\dot{x} = sigma*(y-x)$$
$$\dot{y} = x*(rho - z) - y \tag{2.4}$$
$$\dot{z} = x*y - beta*z$$

where sigma, rho and beta are the controlling parameters. In the figure II-4 these parameters are defined as sigma=10, rho=28 and beta=8/3 and the *Ordinary Differential Equations* in MATLAB has been chosen to solve the differential equations. (With τ =100)

These two figures show that simple equations can generate very complex systems which seems to be entirely random and unpredictable. An Analysis of these time series by means of conventional classic methods could lead to the conclusion that the variation in these two time series are more or less under random fluctuations and therefore, a good estimation of the future behavior would not be possible. Nevertheless, applying the modern methods based on non-linear dynamics and chaos theory can make good predictions, even when the system equations are unknown. Today, such predictions are used as the basis of new techniques in signal processing and noise reduction.

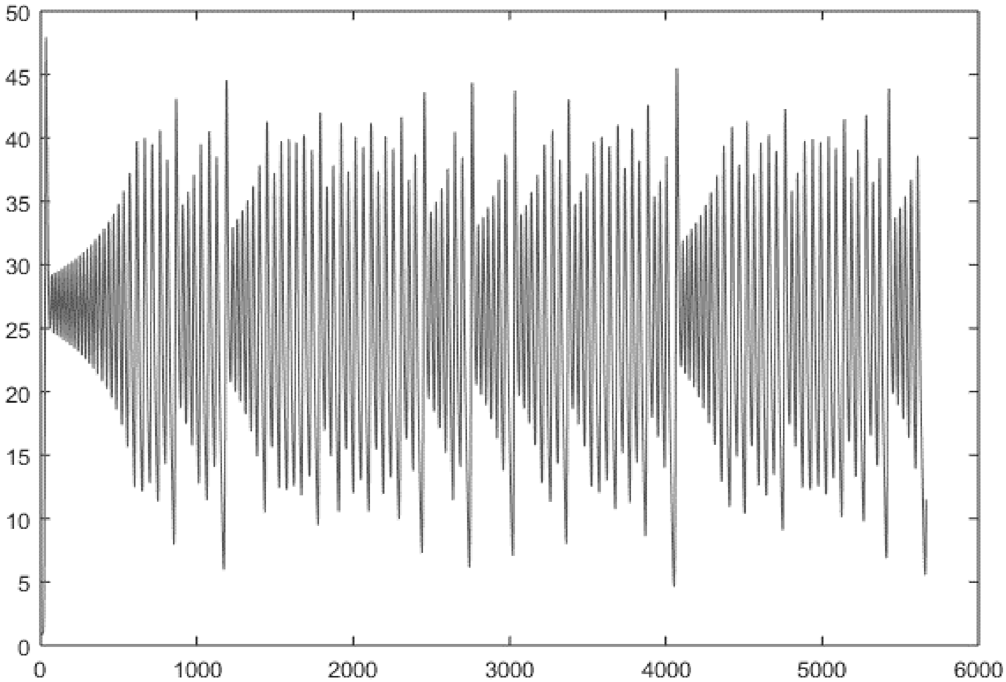


Figure II-4: Time series derived from z co-ordinate observation of the Lorenz Equations (τ =100)

It may seem that the deterministic equations (2.3) and (2.4) cannot cause any uncertainty. This is true, if the initial state of these systems are known, whereas by a slight change in the initial conditions, the evolution of the system and time series can be totally different, because of the chaotic characteristics of them. For instance, in the Henon map, the initial state of (x_0 , y_0) is slightly transferred to $(x_0 + 10^{-10}, y_0 + 10^{-10})$. Figure II-5 shows the observation of x' co-ordinate with the new initial state. It may seem that in this condition the time series is similar to the one in figure II-3. In order to show the difference, the $y_n = x_n - x'_n$ is plotted in figure II-6. It is obvious from this figure that after the first iterations, y_n increases rapidly until there is totally no correlation between the two time series. This means that an initial uncertainty of 10^{-10} has led to a total lack of knowledge on future behavior of the system. Such exponential growth of the difference is one of the characteristics in chaotic systems.

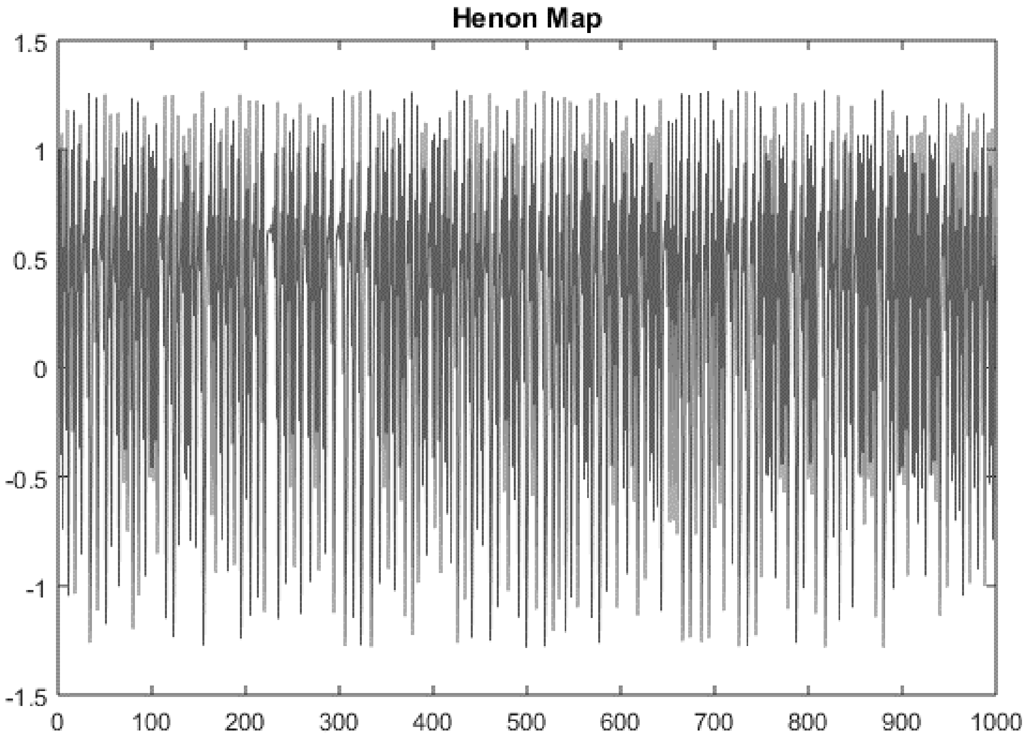


Figure II-5: Observation of x' co-ordinate of Henon map with the new initial state $(x_0 + 10^{-10}, y_0 + 10^{-10})$

Another famous example is the double pendulum. It is a pendulum with another pendulum attached to its end, and is a simple physical system that exhibits rich dynamic behavior with a strong sensitivity to initial conditions. The double pendulum undergoes chaotic motion and the observers of two similar double pendulum systems with a very tiny difference of 0.01 in the initial condition, experience two total different dynamics.

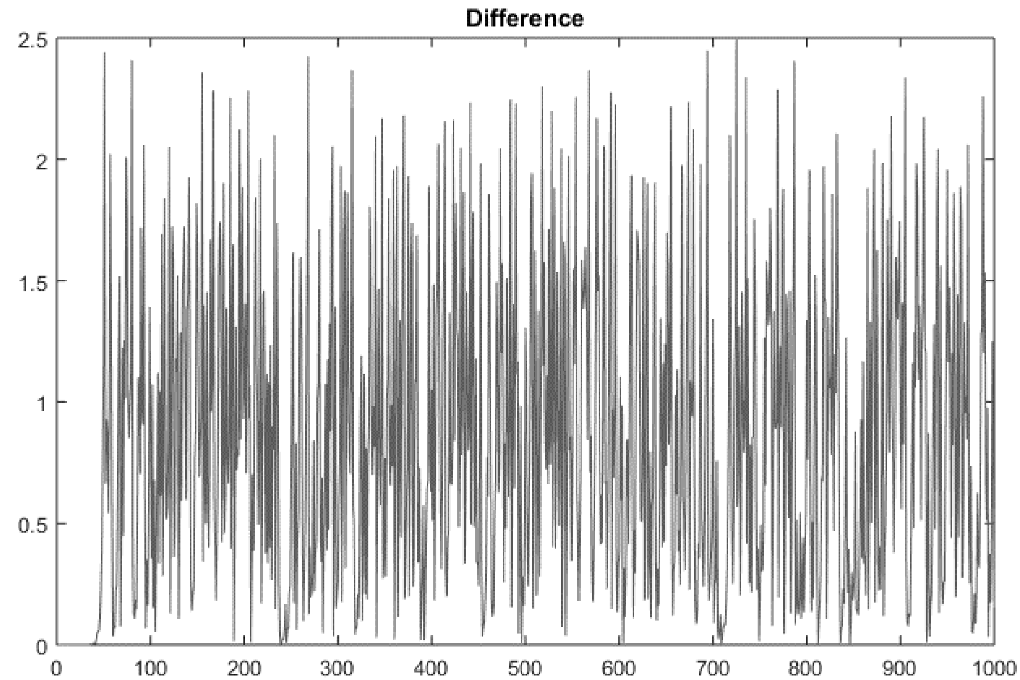


Figure II-6: Difference between two time series derived from Henon map with 10^{-10} difference in initial states

It is also important to note that the predictability of chaotic systems is very dependent on the choice of time scale and sampling interval. Figures II-7 and II-8 show the same time series derived from Lorenz attractor in figure II-4 with different sampling interval. In figure II-7 the τ is equal to 300 and in figure II-8 it is equal to 10. These figures indicate that normally the deterministic nature of the time series is more obvious when smaller time interval are taken into consideration or the sampling is done more frequently. (Stark, 1994) (Yuh-Jye Lee, 2013) On the other hand, a very small time interval may also lead to valuable information loss related to the dynamics of the system. (Abarbanel) (Hilborn, 2000)

Based on this, it can be concluded that for a better electrical load forecasting, it is very important to select the best time interval. In this way, finding the similarities in time series would be much easier which eventually results in better performance of the model.

The presented method for predicting the electrical load in this dissertation is based on an ANFIS-Chaos model. The first step in applying this method, is to determine whether the original data is based on deterministic chaos or not. In the next part three methods will be introduced. The indicators derived from these methods can determine whether the system is chaotic or random. These methods will be applied on the electrical load data of CUTEC to prove the chaotic nature of them.

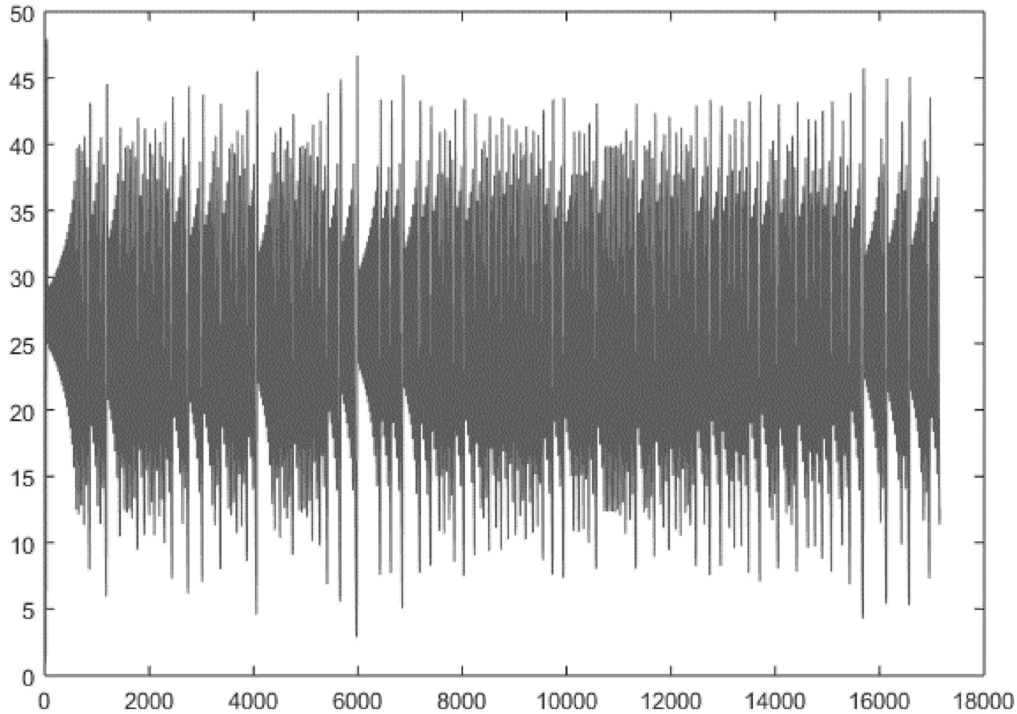


Figure II-7: Time series derived from z co-ordinate observation of the Lorenz Equations (τ =300)

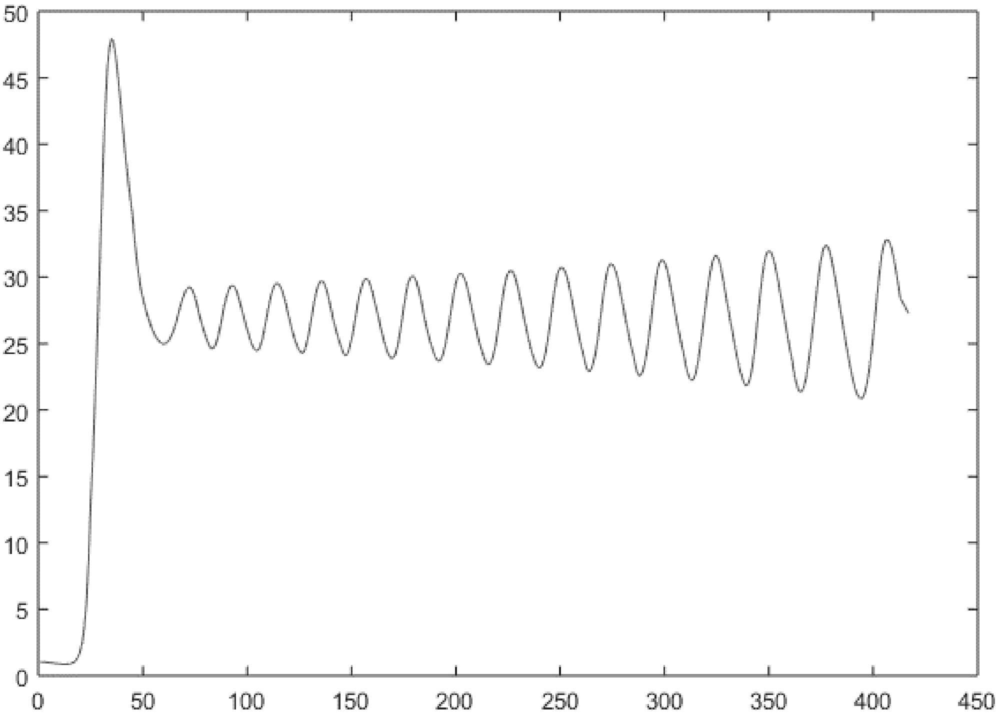


Figure II-8: Time series derived from z co-ordinate observation of the Lorenz Equations (τ =10)

3. Proving the Existence of Chaos in Sample Data

3.1. Embedding Dimension

There are various technics to examine the presence of deterministic chaos in a complex time series. These indicators can distinguish whether the time series is originated from a chaotic system or a random one. Most of these methods are developed for understanding the dynamics of time series. The main idea behind these technics is to take one or more quantities which are constant and invariant under co-ordinate changes into consideration, try to reconstruct the time series based on them and finally examine the presence or absence of chaos in it.

In the last part, the observations of only one co-ordinate (x in Henon map and z in Lorenz equations) of the introduced systems were plotted and shown in the figures. It may seem that the information based on only one co-ordinate time series is not sufficient to reconstruct the function F, when the dynamic map of the system is not known. The *Takens Embedding Theorem*, however, states that under given circumstances for a typical measurement function F, it is possible to reconstruct the state X and the map $x((n -1)\,\tau) \rightarrow x(n\,\tau)$, only based on a limited observed time series of one co-ordinate. In fact, the Takens' embedding theorem tells that the information about the hidden states of a dynamical system can be preserved in a time series output, whilst the original state and dynamics might be inaccessible and the reconstructed dynamics is completely equivalent to those of the original. (Yap, 2011)

On the other hand, as we know, the state of chaotic systems in each moment depends on various parameters and therefore k in the equation (2.2) in part 2.1 is a large number. As an obvious consequence of this, the data volume describing the system increases significantly which makes the prediction process infeasible. There are many different methods to reduce the system dimensions and select the parameters that describe the system behavior with an acceptable error. One of the key methods in this regards is the same *Takens Embedding Theorem*. This method ensures the completely equivalent dynamics of the system, while reducing the system dimension from k to d and considering $y(t)$ instead of $x(t)$:

$$s(t)= h(x(t)): \mathbb{R}^k \rightarrow \mathbb{R}^d \tag{2.5}$$

$$y(t)=[s(t),\ s(t+ T),\ s(t+ 2T),\ \dots ,\ s(t+(d-1)\ T)]$$

Where T is the interval between the observations (or lag) and defined as:

$T= m\ \tau$ and $(m = 1, 2, 3, \dots)$

And d is called the embedding dimension. According to this theorem if the value of d is large enough, $y(t)$ can reconstruct the important dynamics of $x(t)$. These indices can be used to achieve enough information and reconstruct a similar system with the same dynamics, but here we use these parameters as indicators, which can verify, if the original system is chaotic or not.

It is very important to choose the d as small as possible on account of efficiency and numerical accuracy. Choosing the observation interval T, as mentioned in part 3.1, can also be crucial, because it can affect the accuracy of prediction and also the complexity of the time series. (Liebert W., 2014) (Buzug & ., 2008). There are different methods to calculate the best value for T. Here the *Mutual Information function* is used to calculate the observation interval. The mutual information of two variables specifies the amount of information obtained from one random variable through the other one. It shows the rate of mutual information of each elements or a part of the time series with the main one. A time-lagged mutual information diagram is generated by shifting the time series by one element and calculating the mutual information between the new time series and the main one. In this way, it is possible to analyze the time series to find the parts, which have the most similar dynamics. As a matter of fact, the extremums in the time-lagged mutual information diagram are the critical points in time series, which may be a sign of the mentioned concept drift. By calculating these lags, it is possible to categorize the time series based on the most similarities and prepare the datasets for ANFIS training and testing, so that instead of corresponding one input to one output as the training sets, the input will be corresponded to two or more outputs, whose similarity has been proven by the mutual information function.

The *Mutual Information function* is defined as:

$$I(\tau) = \sum\nolimits_{s(n),s(n+\tau)} \Pr(s(t), s(t+T)) \log\left[\frac{\Pr(s(t),s(t+T))}{\Pr(s(t))\ \Pr(\ s(t+T))}\right] \tag{2.6}$$

where Pr is the probability distribution function which is derived from the data histogram. (Simon, 2006) (Osórioa, 2015)

Here, the electricity load of the CUTEC institute in Clausthal-Zellerfeld in Germany in 2014 is taken into consideration as the sample data. This data presents the electricity consumption of this institute with almost 100 employees, several industrial facilities and pilot plants, which results in a sample data with lots of extremums. This makes the load prediction a challenging process. The energy park will be introduced in chapter V part 3.1 in detail. This data has a time resolution of 15 min and will be the foundation for the load prediction algorithm presented in this dissertation. Figure II-9 shows the time-lagged plot of mutual information function for this sample data. (Implemented in MATLAB - Appendix A.1)

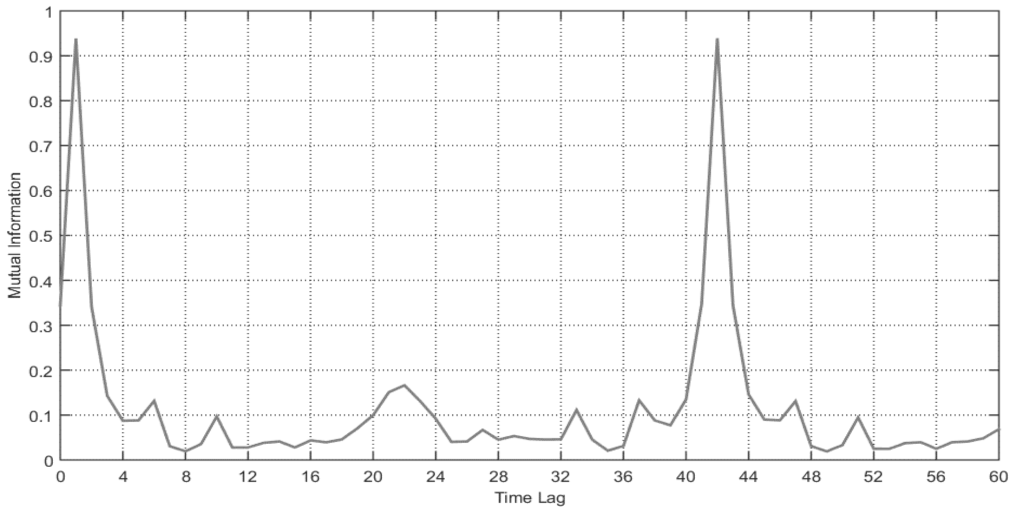


Figure II-9: Time-lagged plot of mutual information function for the sample electical load

The extremums in this diagram show the critical points in different parts of the main time series, such as the points with the highest similarity. These lags are very important for the process of training and should be examined for each prediction time horizon separately (weekly, monthly and seasonal). Selecting the best lag or sets of lags is based on trial and error in the training process. The best lags can be determined based on the best and most accurate results in each prediction period. (See section 4 in chapter 3 for the mutual information diagram for each time horizon)

The importance of determining an appropriate minimum embedding dimension d is discussed before. Here, the method of Cao is used. This method has several advantages such as:

- It can clearly distinguish a deterministic time series from a stochastic one.
- It does not depend on the number of available data points.
- It can also be applied for times series with higher dimensions.

- It is numerically accurate and computationally efficient.

This method uses the following equation to calculate *d*:

$$E1(d) = \frac{E(d+1)}{E(d)} \tag{2.7}$$

where *E(d)* is:

$$E(d) = \frac{1}{N-dT}\sum_{t=0}^{N-dT-1} \frac{||y_{d+1}(t)-y_{d+1}^{NN}(t)||}{||y_d(t)-y_d^{NN}(t)||} \tag{2.8}$$

In this equation *N* is the length of the time series, *d* is the embedding dimension and NN stands for the nearest neighbor. ||.|| is some measurement of Euclidian distance and is given here by the maximum norm, i.e.

$$||y_d(t) - y_d^{NN}(t)|| = \max_{0\le j\le d-1}[s(t+jT) - S^{NN}(t+jT)] \tag{2.9}$$

In the equation (2.8), $y_{d+1}(t)$ is the reconstructed vector with embedding dimension *(d+1)*. If *d* is qualified as an embedding dimension, then any two points which are close in the *d*-dimensional time series, will be still close after the reconstruction in *(d+1)* dimension.

In equation (2.8), *E(d)* is dependent only on two parameters: embedding dimension *d* and time interval *T*. Equation (2.7) investigates the variation of *E(d)* from *d* to *(d+1)*. As the value of *d* increase, *E1(d)* gets closer to one. The best value of *d* is when the *E1(d)* stops changing.

It should be noted that the value of *T* must be given before determining the minimum embedding dimensions. Different *T* can lead to different values for minimum embedding dimensions.

As mentioned before, it is possible to distinguish deterministic time series from random time series with this method. Let:

$$E^*(d) = \frac{1}{N-dT}\sum_{t=0}^{N-dT-1} |s(t+dT) - s^{NN}(\mathrm{t}+\mathrm{dT})| \tag{2.10}$$

where $s^{NN}(\mathrm{t}+\mathrm{dT})$ is the nearest neighbor of $s(t+dT)$. A new parameter can be defined as below:

$$E2(d) = \frac{E^*(d+1)}{E^*(d)} \tag{2.11}$$

For time series from random set of numbers, *E1(d)* will never reach a saturation as *d* increases, but sometimes in observations, it is not obvious whether *E1(d)* changes slowly or it has stopped changing. [3] Another sign of a random time series is when the index *E2 (d)* will be equal to one for all the data, since the future values are totally independent of the past values. In deterministic data, *E2 (d)* is related to *d* and therefore, it will not be constant for all the data. In chaotic time series, the value of *E2 (d)* is always less than one for small values of *d*. (Cao, 1997)

Figure II-10 shows the graphical illustration of *E1 (d)* and *E2 (d)* values from the equation (2.10) and (2.11) for the electricity load data of CUTEC institute which is derived from the code written for this function in MATLAB. (Appendix A.2) According to the figure, the value of *E1 (d)* increases to a certain point (*d*=3) and then the changes are very slow implying that the embedding dimension in this time series is equal to 3. It is very important to note that the value of *E2 (d)* stays less than one for small values of *d* which implies the chaotic nature of load data.

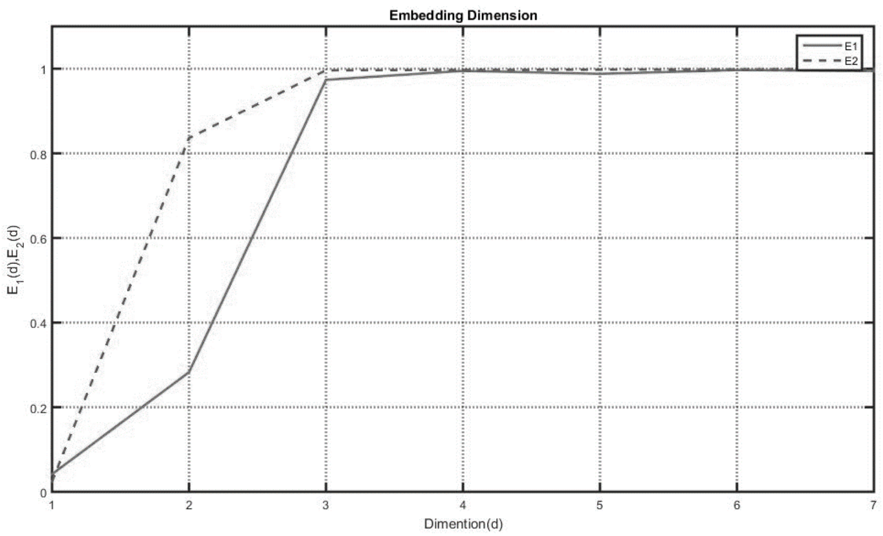


Figure II-10: E1 (d) and E2 (d) graphs for electrical load data of CUTEC institute in 2014 (for T=8 and d=7)

The above figure was the first clue to the existence of chaos in the sample time series.

3.2. Grassberger-Procaccia Correlation Dimension

The geometric language of chaos is Fractal. This concept was developed specifically in physics and mathematics by Hausdroff in 1919 and was broadly developed by Mandelbrot in 1967 to describe the irregular and random phenomena in nature. He attracted significant attention in 1983 by proposing the fractal geometry as the underlying geometry in nature. (Mandelbrot,

[3] In fact, if the sample data is very limited, it may happen that *E1 (d)* stops changing, although the time series is random.

1982) The book *"The Fractal Geometry of Nature"* became a classic of chaos theory and this concept have had significant importance in describing chaos phenomena since then.

In Euclidean geometry, objects are not Fractal, they are comprised of lines, rectangular volumes, arcs, cylinders, etc. which are classified as belonging to an integer dimension (either 1, 2, or 3), while in a fractal pattern, the repetition of the structure causes a non-integer dimension. In fact, a fundamental parameter in describing a fractal pattern is its dimension (*D*). This parameter is one of the invariant quantity under co-ordinate change in time series.

Various algorithms have been introduced in recent years to calculate the fractal dimension among which, the *correlation dimension* is the most famous one due to quick calculations and being less affected by noise, even when small number of points are available. This technic was introduced in 1983 by Grassberger and Procaccia. It is a measure of the dimensionality of the space occupied by a set of given data. For example, if all the random-selected points are identical, then the correlation dimension (D_C) will be zero. If they lie on a smooth curve, (D_C) will be equal to 1 and if they are distributed on a triangle embedded in three dimensional space the (D_C) will be 2. The main application of the Grassberger-Procaccia algorithm is however, to detect the possible fractional dimensions of a strange attractor.

Now for a given *R*, let *N* be the number of pairs (*i,j*) in some metric space with distance $|s(i)-s(j)|$ between any pair of points. The correlation sum *C(R)* is then defined as the fraction of pairs which have a distance between them that is smaller than *R*:

$$C(R)=\frac{1}{N(N-1)}\sum_{i=0}^{(N-dT-1)}\sum_{j=0\,,i\neq j}^{(N-dT-1)}\varphi(R-|s(i)-s(j)|) \tag{2.12}$$

where $\varphi(x)$ is the Heaviside step function and defined as:

$$\varphi(x)=\begin{cases}0, & x<0\\ 1, & x\geq 0\end{cases} \tag{2.13}$$

As the number of points in the equation (2.12) tends to infinity, and the distance between them tends to zero, the correlation integral *C(R)* for small values of *R*, will take the form:

$$C(R)\sim R^{Dc}$$

which means:

$$C(R)=\lim_{R\to\infty} R^{Dc} \tag{2.14}$$

and as a result, the correlation dimension will be calculated by:

$$D_C = \lim_{R \to \infty} \frac{\log C(R)}{\log R} \tag{2.15}$$

The simplest way to estimate D_C is to plot $\log C(R)$ against $\log R$ and to fit a straight line to the small R tail of the curve. D_C is then the slope of this line.

When D_C is not an integer, it implies that the system contains a strange attractor. This is usually a sign of chaos. (Grassberger, 1994) (Grassberger P. a., 1991)

The Grassberger-Procaccia algorithm was applied on the empirical electrical load data of CUTEC in 2014, by writing a code in MATLAB. (Appendix A.3) To estimate the correlation dimension in this data, $\log C(R)$ is plotted against the $\log R$ in figure II-11. The slope of the linear part of this graph, shows the D_C with a very good approximation. The value is equal to 6.0837 which is a non-integer, implying the chaotic behavior in the sample load data.

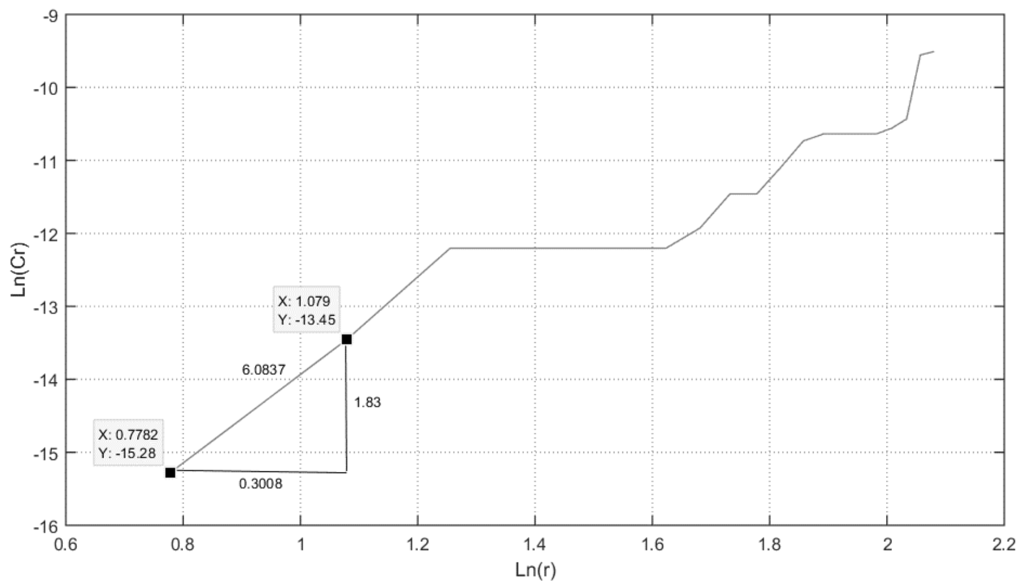


Figure II-11: Correlation dimension in CUTEC load data based on the Grassberger-Procaccia algorithm

3.3. Lyapunov Exponent

In mathematics, the Lyapunov exponent or Lyapunov characteristic exponent of a dynamical system shows the rate of divergence of nearby trajectories- a key element in chaotic dynamics. Lyapunov exponent is an indication to show a chaotic system response to small perturbations

For one dimensional maps the exponent is simply the average log $\left|\frac{df}{dx}\right|$ over the dynamics. For higher dimensional maps, the rate of separation can be different for different orientations of initial set of points. In fact, the number of exponents is equal to the dimension of the phase space. Therefore, there is a *spectrum of Lyapunov exponents*. Generally, the largest one is referred to as the *maximal Lyapunov exponent* or λ_{max} which is an indicator of predictability of the system. This index is defined as:

$$\lambda_{max} = \frac{1}{N\Delta t}\sum_{t=0}^{N-1} \ln\left[\frac{[s(t+\Delta T)-s\prime(t+\Delta T)]}{[s(t)-s\prime(t)]}\right] \tag{2.16}$$

here, N is the time series length, $s(t)$ is the reconstructed scalar function based on the embedding dimension, $s'(t)$ is the scalar function adjacent to $s(t)$ and Δt is the time interval.

A positive Lyapunov exponent may imply the existence of chaos in the system. For the attractors of maps, the Lyapunov exponents can also distinguish between different types of dynamics. For instance, a fixed point has only negative exponents or a limited cycle has one zero exponent and the rest of the spectrum is negative. (Cencini & al, 2010) (Xun-yi Ren, 2012)

The sample load data was tested with the Lyapunov exponents algorithm and the result from the MATLAB implementation is presented in the figure II-12: (Appendix A.4)

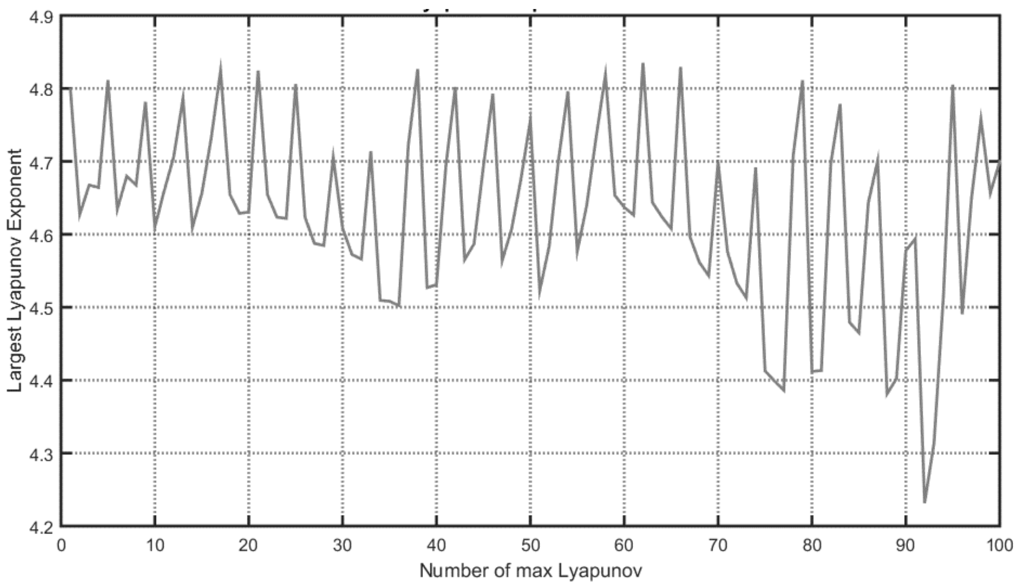


Figure II-12: Lyapunov exponents for electrical load data of CUTEC implying the chaos in the time series

The diagram shows that the Lyapunov exponents are all positive, which is a clear sign of chaotic behavior in the studied time series.

Summary of the chaos indicators:

In this part the electrical load data of CUTEC in 2014 was tested with three different methods in chaos detection, namely the embedding dimension, the Grassberger-Procaccia algorithm and the Lyapunov exponents. The first clue for chaos in the selected time series was the fact that the value of $E1(d)$ reached a saturation and value of $E2(d)$ was less than one for small values of d in the embedding dimension method. Also, the slope of the correlation dimension graph in figure II-11 is a non-integer which shows the fractal dimension of the time series and implies the chaotic behavior of the data. And finally, the positive Lyapunov exponents' spectrum in figure II-12 proves the existence of chaos in the electrical load data.

These indications state that the considered time series is originated from a deterministic system and it is predictable based on just the time series itself. In the next part the flowchart of the proposed ANFIS-Chaos model for the prediction of electrical load data and different steps will be presented.

4.Proposed Algorithm

This dissertation proposes a new approach in load forecasting, which uses the Adaptive Neuro Fuzzy Inference system (ANFIS) for a systematic training and testing the model by using the concept drift for categorizing the input data. All the related functions for this model are implemented in MATLAB. The basic structure of this model was presented in the beginning of this chapter. Figure II-13 shows the detailed flowchart of the proposed algorithm and the ANFIS-Chaos model.

The left part of this flowchart contains the algorithms which were discussed in this chapter.

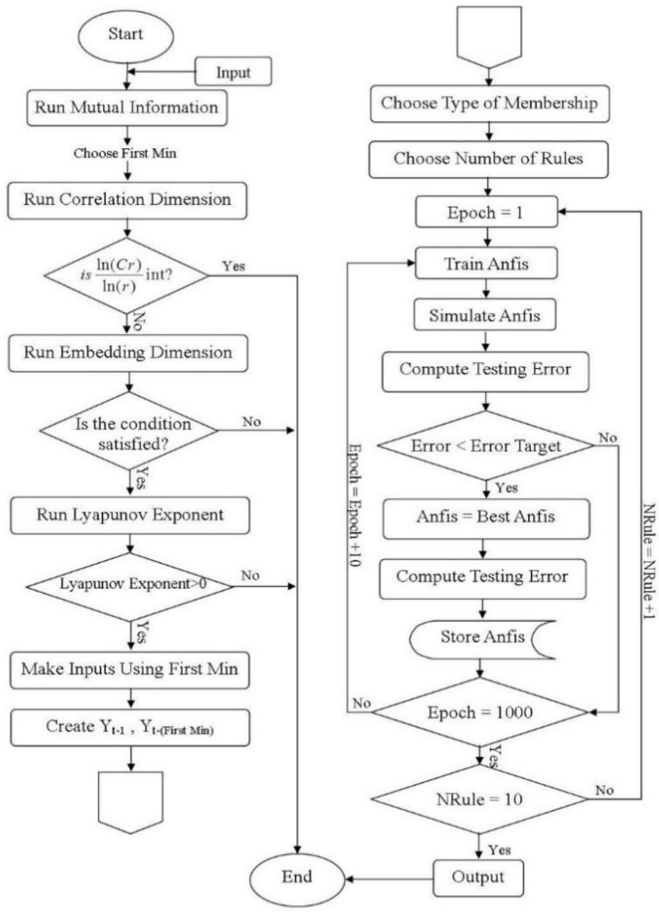


Figure II-13: ANFIS-Chaos Model for prediction of the electrical load

The Steps of the proposed algorithm

1. Entering the empirical load data into the model
2. Running the Mutual Information function (Concept drift)
3. Determining the best lags for the final training process (trial and error)
4. Running the Correlation Dimension algorithm and measuring the dimensions of the system by using some random points.
5. According to the results in step 4, if $\frac{\ln(C(R))}{\ln(R)}$ is a positive integer, the system is not chaotic and the algorithm will be terminated. Otherwise, the process will continue. It should be noted that here the value of R is very small and as shown in figure II-11 the slope of the linear part of the graph should be considered.
6. Based on the results from step 5, if the existence of chaos in system is granted, the process will continue by applying the Embedding Dimension algorithm based on the

equations (2.5) to (2.11). At this point if the existence of chaos in system was confirmed, the main algorithm will continue, otherwise, it stops.

7. If the system passes the step 6, it will be tested in this step by the Lyapunov exponents algorithm (equation (2.16)). This step investigates the chaotic behavior of the system based on the degree of convergence or divergence. If the maximum Lyapunov exponent is larger than zero, the system has a chaotic nature.

Passing all these steps proves that the system is definitely chaotic and the algorithm continues to the final steps. Here the input data will be prepared based on the lags (step 2), and fed into the ANFIS model systematically. It should be noted again, that the first extremums of mutual information are not necessarily the best options for the lags and through a trial and error process the best lags will be selected based on the results they provide in the training and testing process.

In the next chapter, the idea of electricity load prediction in general will be introduced and discussed. Then the forecasting methods using simple neural networks will be reviewed in detail.

III. Chapter 3

In this chapter the importance of electrical load prediction in today's energy systems will be discussed and the historical development of methods to achieve accurate and efficient models in load forecasting will be studied. The recent approaches toward short and long-term load forecasting in other studies and articles will then be reviewed. Introducing the Artificial Neural Networks (ANNs) as one of the most common and popular methods in load forecasting, the basics and structures of the ANNs will be explained in details and the advantages and disadvantages of this method will be discussed. Finally, the sample electrical load data from CUTEC in 2014 which is already prepared based on the lags from the mutual information function, will be used to train a simple ANN system and the results of the load prediction based on the trained system will be given to create a basis for a final comparison between the results of simple ANN method and the proposed algorithm in this work.

1. Load Prediction in Today's Energy Systems

Precise electricity consumption forecast plays a vital role in the modern energy planning and has primary importance in energy system design, modellings and simulations, which entail an investment of billions of dollars. An accurate electrical load prediction is an essential part of an efficient planning and operation of electrical power systems from generation to design, expansion plans and construction of electricity transmission network and finally to the distribution facilities. This makes the electrical load forecasting very important not only for the governments and public organizations, but also for the private sectors, investors and companies.

The quality control of electrical power systems, as well as production management, transport and distribution of that energy is very dependent on the accuracy of the load forecast. Sufficient electricity generation capacity needs an accurate forecasts of the electrical consumption and also the timing of future demands. An overestimation of the future load may lead to unnecessary developments in electricity generation, which means wasted financial resources. On the other hand, underestimation of the future consumption can cause troubles in energy supplying and higher operational costs.

Based on the application, the forecast horizon may change from very short-term to very long-term. Basically, a short-term load prediction has to be performed to adapt the electricity generation to the electricity consumptions, while a long-term load forecasting is necessary for the energy investment planning. Table III-1 reviews the main applications of load forecasting based on the prediction horizon: (Weron, 2007)

Table III-1: Different types of load prediction

Type of prediction	Time of Prediction	Application
Very Short-term	A few seconds to a few minutes	Production adaption Distribution timing Risk analysis for the system security
Short-term	One half to one hour	Storage Operational planning Grid connection planning of the units Maintenance and overhaul scheduling
Mid Term	A few days to a few weeks	Planning for the seasonal peak load (summer and winter)
Long-term	A few months to a few years	Planning the network expansion and production growth

As shown in the table, the information required for the optimal planning in the electrical power systems will be available based on different types of load forecasts.

In general, the load forecasting methods can be divided into two main categories: The traditional methods and the modern ones. The traditional methods, mainly, include statistical analysis, mathematical interpolation and regression process, while the modern approaches deal with the load prediction based on artificial intelligence.

The traditional methods lose their accuracy, when rapid changes occur in the input data, for instance when a sudden change happens in the weather and meteorological data, while the new methods improve the accuracy of the prediction by making nonlinear models between the input and output variables, therefore the modern methods are more efficient and have more acceptance in today's network studies. (Park, 1991)

Table III-2 shows the application frequency of ANNs as one of the most popular method in modern approaches toward different problems in different study fields in electrical power systems. (Islam S. M.-A., 2001)

Table III-2: The application rate of ANNs in power systems

1	Load forecasting	22%
2	Fault diagnosis	18%
3	Transient stability	14%
4	Dynamic security assessment	12%
5	Warning	10%
6	Identification and control	9%
7	Operational planning	7%
8	Static security assessment	5%

As shown in the table, the most frequent application of the ANN- based methods are in load forecasting. Table III-3 reviews the rate of application of this method in different types of load prediction: (Danladi, 2016)

Table III-3: The application rate of neural networks in different types of prediction

1	Short-term	62%
2	Mid term	14%
3	Long-term	24%

However, the artificial neural networks by itself cannot ensure the best prediction results. There are some drawbacks in using ANNs for prediction which will be discussed in the part 5.

The next part tries to review the development process of different methods and approaches toward the problem of electrical load forecasting during the past years.

2. Load Prediction Methods

As mentioned before, a precise electrical load forecast has always been of great interest in energy system analysis and therefore there has been many efforts to achieve models which can predict the future electrical load data. Since many years ago, the electricity demand forecasting has become one of the major research fields in electrical engineering.

For many years the forecasting was done based on the billing record and weekly historical data, which were collected and presented graphically to the responsible people to forecast the future load based on their many years' experience. (Medha Joshi, 2017)

But probably one of the first studies in electrical load forecasting was done by Heinemann *et al.* in 1966 which studied the relation between load and temperature. (Heinemann, 1966) Additionally, an early survey on load forecasting techniques has been introduced in 1968 by Matthewman and Nicholson. Lijsen and Rosing (1971), Meslier (1978), Abu El-Magd and Sinha (1982), Bunn and Farmer (1985), and Gross and Galiana (1987), Moghram and Rahman (1989) are just a few scientist who studied the load demand modelling and forecasting in 70's, 80's and 90's. (Medha Joshi, 2017) The methods introduced during this period and the following years vary widely in their complexity, applicability, precision and sensitivity, but they can mainly be categorized into two groups: the classic methods and modern approaches.

2.1. Classic Methods

The traditional methods for electrical load forecasting are usually based on statistics and tries to find a linear relation between the electrical load and several input parameters. (Heiko Hahn, 2009) In the following parts, the most famous classic methods will be introduced.

2.1.1. Regression Algorithms

This method is one of the most used models for electrical load forecasting. Regression methods try to model the relationship between the influencing factors such as weather condition, day type, etc. and the electricity load pattern by defining a mathematical equation. For example the simple regression method models this relation as a simple straight line, while multiple linear regression (MLR) expresses the future electrical load pattern as a function of several explanatory variables. The regression based algorithms are of high complexity, with heavy computational load and extensive computational time. (Islam E. B., 2011)

2.1.2. Classic Time Series Analysis

The time series analysis is another widely used method in load forecasting. In its classic form, this method predicts the future state of electrical load by finding a linear combination of previously recorded value and extrapolation of that combination to the future. The most popular classic time series analysis methods are ARMA and ARIMA introduced by Box and Jenkins

(Hong W. , 2013). Both of these methods are composed of an autoregressive (AR) part and a moving average (MA) part. The moving average refer to the process of calculating a new average, when a new value of observation become available. A developed version of these methods with exogenous variables is called ARIMAX which is the most famous tools among other classic methods based on time series analysis. This method claims that the load generally depend on the time and weather conditions. However, these methods show very inaccurate results, since they are based on simple mathematical modeling. Moreover, the large space needed for storing all the previous observed values is another weakness of these methods. (Abdullah S. Al-Fuhaid, 2009) (Reddy, 2008) (Gabriela Grmanova, 2016)

2.1.3. Exponential Smoothing

The Holt-Winters exponential smoothing method attempts to predict the future load by smoothing empirical observations in order to decrease the inaccuracy caused by randomness involved in the system. Unlike the classic time series analysis methods, here the historical observations are not weighted equally and the older observations' weights decrease exponentially with time. Consequently, the newer data has stronger influence on the model. Mathematically, this method is represented as follows:

$$Y_{t+1} = \alpha X_t + (1-\alpha) Y_t \tag{3.1}$$

Where Y_{t+1} is the forecast value at time t+1 and Y_t , X_t are the actual value at time t. In this equation, the coefficient α is defined as: $\alpha = 1/N$, where N is the number of observations included in the average. (Winters, 1960) (Holt, 1967)

2.1.4. Adaptive Filtering

In this method the difference between the actual value and the predicted value will be calculated, when a new observation is available. According to this difference, a set of weights will be assigned to each observation and the future value of load will be calculated based on the following formula:

$$Y_{t+1} = \sum_{i=1}^{N} w_i \ X_{t-i+1} \tag{3.2}$$

Where Y_{t+1} is the prediction value at *t+1* and X_{t-i+1} is the observation value at *t-i+1*. Here, w_i are the weights of each observation and they are calculated as follows:

$$Y_{t+1} = \sum_{i=1}^{N} w_i \ X_{t-i+1} \tag{3.3}$$

The Adaptive Filtering (AF) method is a repetitive process, meaning that the procedure is repeated until the best weights for the observations are achieved. Although there has been reports with very good results on load forecasting with this method, they are only valid under the very special and rare condition that the load data trend is linear. (Park & Y. M. Park, 1991) (Igor R. Krcmar, 2010)

2.1.5. Similar Day Lookup

This technique is based on the idea that the similar days is the best source for the prediction of the days with almost the same conditions. This similarity can be the weather conditions, temperature, humidity, day of the week and etc. In order to make the prediction process more accurate, the difference between different influencing parameters will be weighted and considered in the prediction process. Obviously, the more historical data available, the more accurate will the load forecasting be. (Medha Joshi, 2017)

2.2. Modern Approaches

The mentioned methods and techniques in part 2.1 are basically applicable only under the condition that the load data is linear. The results for prediction of a non-linear load with these methods are far from accurate. By the introduction of artificial intelligence (AI) in the past years, there has been a huge development in the electrical load forecasting techniques. These new approaches are not limited to linear load data and they show relatively good results also in non-linear systems.

In the followings, some of the most popular modern approaches will be introduced and the advantages and disadvantages of them will be discussed briefly.

2.2.1. Fuzzy Logic

Fuzzy logic has been widely used in power system problems especially in load forecasting methods. It is based on the usual classic Boolean logic, but it is more powerful and effective in comparison to the classic methods. By using the Fuzzy logic, the outputs can be derived from the inputs logically based on the defined "If –Then" rules. The main parts of a system based on this method are:

- Fuzzification, which classify the input based on the membership functions
- Fuzzy Inference, which evaluates the input and create the results based on the defined rules; and finally
- Deffuzification, which changes the created data from the fuzzy state to the real data state.

The advantage of the fuzzy logic-based methods is their simplicity and the manipulability of the procedure and rules. However, the lack of a self-learning intelligence is the main drawback for this approach. (Pandian SC, 2006) (unkas, 2015) (Chi Xie, 2006)

2.2.2. Support Vector Machines (SVM)

Support Vector Machine or shortly SVM is a machine learning method. It is a very promising and highly principled technique for data classification and regression. The SVM has a non-linear part and therefore it can map the non-linear relation between the output and the input. Since the introduction of SVM as a method in load forecasting in 2001, there has been many studies based on this technique. Most of them use Kernel functions in order to map the data into a high dimensional system. In recent years, there has been more developments in the SVM-based methods. For instance, in 2007, Gang Li *et al.* introduced the SCE-UA algorithm (Shuffled Complex Evolution-university of Arizona) in SVM method and in 2015 Fazil Kaytez *et al.* developed the Least Squares Support Vector Machine (LS-SVM). The results in both studies showed even better results in short-term load forecasting in comparison to simple Artificial Neural Network approach. (Gang Li, 2007) (Fazil Kayteza, 2015)

However, it should be mentioned that in most cases, some other influencing factors are also included in the process, such as the estimation of future temperature. Including unprecise information in the procedure causes higher uncertainty on prediction and may lead to higher error rates.

2.2.3. Artificial Neural Networks (ANN)

The Artificial Neural Networks (ANNs) have received the highest share of interest in the field of electrical load forecasting. There are hundreds of articles and studies based on ANN with the aim to achieve the best forecasting results. The ANNs were introduced in 1960's but their first application in load forecasting is reported in the early 1990's.

ANN is a very efficient and high classed processing system with an effective and principled specifications in non-linear modeling and adaptation. The ANN has the unique feature of self-learning, meaning that the non-linear relation between the output and input data can be learned by training the hidden neuron layers in the system from the past data without any complex dependency assumption between the input variables.

The main drawback in ANN-based methods is their black box nature. When an ANN system is trained based on the input data, the relation between the neurons is not available and it is not possible to manipulate the mathematical relation in the hidden layers. Also, as mentioned in chapter 1, most of the methods using ANN as their main procedure, take some input variables for the training and prediction process into consideration, which contain unprecise information, e.g. the temperature or humidity prediction for future. Using such information only, may lead to neglecting a great part of other influencing factors and information which can make the prediction more accurate.

2.2.4. Genetic Algorithm (GA)

A very recent technique in load forecasting is developing the neural network methods with optimization processes in order to achieve the optimum structure and weights in the processing system. One of the most successful optimization processes applied is the Genetic Algorithm (GA) which has been studied in several reports and very good results have been proposed. (Metaxiotis & A. Kagiannas, 2003)

Combining two or more of the mentioned methods is very common in order to achieve better prediction results. These hybrid approaches normally overcome some of the mentioned drawbacks in each method. In this dissertation a hybrid method of ANN and Fuzzy Logic will be used for the main process of training and prediction. In this way, the high quality of ANN in self-learning will be combined with the ability of Fuzzy Logic method in defining IF-Then rules to overcome the black box nature of ANN and so the overall system is more flexible. This Hybrid approach together with the classification of the input data based on the concept drift theory and the systematic feeding of these datasets into the ANFIS training system introduces a unique method which presents encouraging results with very high accuracy in short, mid and long-term load forecasting in comparison to the other methods with similar conditions and time horizons.

In the next part an introduction to the Neural Networks in general – as the main idea behind the proposed method – will be given and the training process inside an ANN will be explained. Finally the prepared datasets will be used to perform a prediction with simple ANN to show the importance of the categorization of input data to make datasets for the systematic feeding in the training process and also to make a basis for comparison between simple ANN and ANFIS which will be presented in the next chapter.

3. Neural Networks

3.1. Introduction

Neural Nets are networks consisting of neuro-cells in the brain of human beings and animals. A human being's brain contains 10 to 100 billion neurons. The connections between these neurons forms a network which is the source of all the capabilities and intelligence in the human beings and probably the main reason of the human beings' survival and evolution by adjusting them to the changes in the environmental conditions through the history.

The function of human brain has been a very interesting topic for the medical doctors, biologists, psychologist and probably many other experts. It was the beginning of the twentieth century when the first studies on brain neural networks and the complex connections between the neurons were proposed and proved that all the intelligent abilities of human beings, such as learning, socializing, thinking, etc., are based on these complicated networks.

Since that time the scientist started to model the biological neural networks with mathematical models. The first successful modelling of the brain neural networks was introduced in 1943 by McCulloch and Pitts. (Anderson & Rosenfeld, 1988) However modelling such a complicated system is not the intention of most of the scientist in other fields. They have tried to make mathematical models based on the function of this network and informational connection between the neurons inside it and finally realize it in a physical way. This led to the introduction of the Artificial Neural Networks, which will be reviewed in details in the next part.

3.2. Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) have been inspired and developed based on the human body’s nervous system. In general, a biological neural network is composed of a set or sets of neurons which are physically or functionally interconnected. Each neuron can be connected to a large number of other neurons. These connections are called synapse and they are usually formed from two main parts, namely axons and dendrites. (See Figure III.1) The dendrites make local connections and pass the information from other neurons to the current neuron. The axon can be in form of a nerve fiber through the body and it can be up to one meter long.

As shown in figure III.1, the cell body of the neurons is basically a storage for a very small electrical voltage which has the same function as a capacitor or a small battery. This storage will be charged by the voltage impulse from the other neurons. The more impulses a neuron receive, the higher will be the voltage inside the storage. If the voltage level exceeds a certain threshold in the storage, it will discharge itself by sending a spike via axons and the synapses. The stream will then be split and reaches many other connected neurons through their synapses and this process goes on.

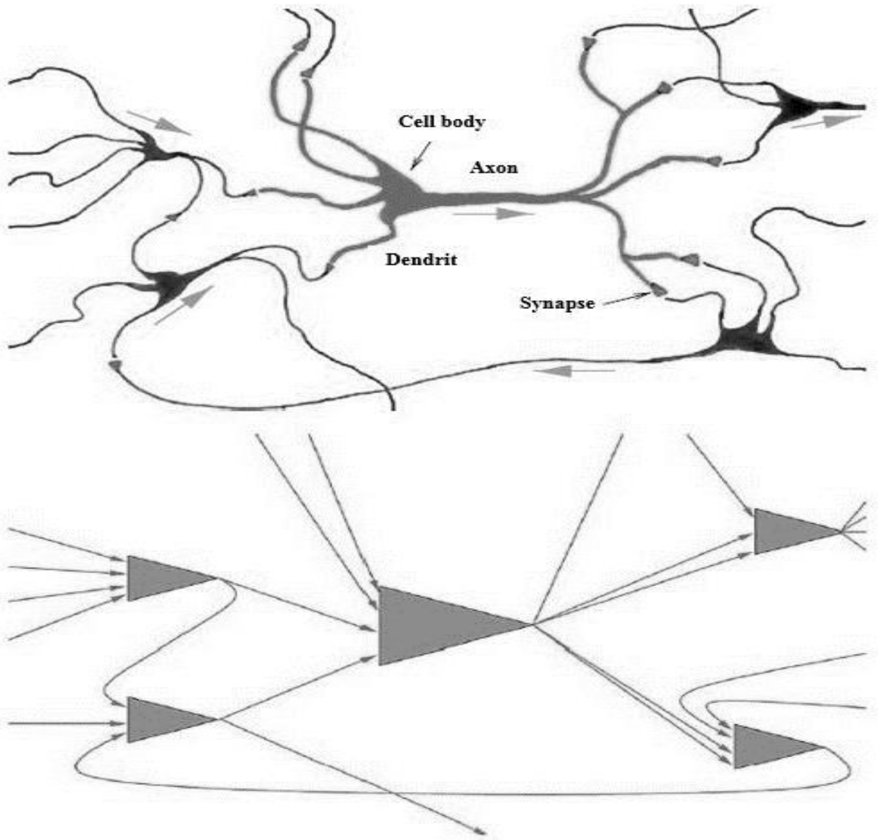


Figure III-1: A biological schematic of simple neural networks (Ertel, 2016)

The surprising fact about this system is that the learning process in the neural networks is not done by the main units (neurons), but it is done by the connections between the neurons (synapses) which have excellent adaptive and flexible characteristics.

As mentioned before, the ANNs are modelled based on the real biological neural networks. These models normally consist of several elements which are called neurons. Each of these neurons is associated with other ones. In this way a network, similar to what we see in human body will be formed. The connection between the neurons in an ANN is weighted. In human brain the connections, which lead to a correct response, are weighted higher and vice versa. Accordingly, in ANN the weights are the parameters which can control the outputs based on the provided input to the system. The weights are specified in the training process and a better training can lead to a better and more accurate outputs.

A neural network may have one or more layers in which two or more neurons can get combined together. These combinations will be connected to the other ones and work in parallel to solve a problem. One of the most important application of the ANNs is in non-linear classification of the input data. The self-learning process in ANN is immune to errors in the training data and this makes ANN the perfect candidate for special processes such as voice recognition, picture recognition and analyzing, Robotic learning, etc. Generally, a neural network has the following capabilities:

- Calculation of a specific function
- Approximation of an unknown function
- Pattern recognition
- Signal processing

In such a network, a data structure will be designed with the help of a programming language. This data structure acts like a neuron and is called a node. The nodes will then be connected together and a learning algorithm will be applied to them. In this way the network can be trained. In an artificial neural network, the nodes can have two states: Active (On or 1) and Inactive (Off or 0). As mentioned before, the connections between the nodes (Synapse) are weighted. A positive-weighted synapse stimulates or activate the next inactive node. The synapse with negative weights change the next connected node’s status to inactive (if they were active before).

The ANNs with their perfect ability to deduce the results from very complex input data can be applied to extract patterns and identify different trends which are very difficult for human or simple computer methods to identify. The main advantages of ANNS are listed below:

- Adaptive learning: The ability to learn how to perform tasks based on the provided information to the system or initial experiences and conditions; it is basically a network correction process.
- Self-organizing: The ANNs can automatically organize and represent the data which have been received during the training process. In fact, the neurons get adapted to the learning rules and response accordingly to the input changes.
- Real-time operators: Artificial neural network computing processes can be done in parallel and with special hardware designed and constructed to obtain optimal results from them.
- Error tolerance: The failures in system caused by errors may reduce the performance of the whole network. However, the main features and capabilities of ANNs will stay the same or very low affected and the system stays robust.
- Classification: Neural networks are able to categorize inputs to get the appropriate output. This feature is especially important for non-linear applications.
- Generalization: This ability enables the ANNs to obtain a general learning rule by dealing with even a limited number of input data and generalize this rule for other observations. In absence of such ability, the system needs to remember infinite numbers of observations, functions and relations.
- Flexibility: A neural network is stable enough to keep its already captured information. At the same time it is so flexible, that it can adapt itself to the new rules and changes without losing any previous information.

The ANNs differ from a traditional computer processes in the following cases:

- The neural networks do not execute the commands in series. They work in parallel and the do not contain any extra memory for storing data and instructions.
- They process the input data in parallel.
- They deal with transformation and mappings rather than algorithms and methods.
- They do not include any complex computational tools; they consists of number of simple, normally weighted, tools.

As mentioned, the ANNs have a different approach toward solving the problems. In traditional computers, the algorithms are applied to solve a problem. These algorithms consist of several solid rules and instructions which would be transformed to simple machine languages, so they could be understandable for the system. However, if the system encounters a problem without any pre-defined specific algorithm, it is not able to solve it. This normally happens when there is no background and records of the similar problems. Hence, the computers can be much more useful if there is a way to analyze the input data based on transformations and mappings even when there is no record of similar problem or there is no pre-defined algorithm for a specific problem. This is where the ANNs can help the computational processes with such problems. In some cases the traditional algorithm processes are still more efficient and provide better results, where in most non-linear cases the artificial neural networks are the better options. Of course the combination of these two can lead to the most optimal way in processing the input data. (Ertel, 2016) (Negnevitsky, 2011)

3.3. Neural Networks Structure

The simplest possible neural network comprises only a single neuron. (Figure III-2) This neuron takes X_1, X_2 and X_3 as input and also a "+1" as intercept term or bias unit. It will then create the non-linear hypothetical output $h_{w,b}$ (x):

$$h_{w,b\ (x)} = f(\,W^T X) = f\left(\sum_{i=1}^{3} W_i\, X_i + b)\right) \tag{3.4}$$

Where W_i is the weight associated to each connection and f is the so called ***activation function***, which limits the output of a neuron.

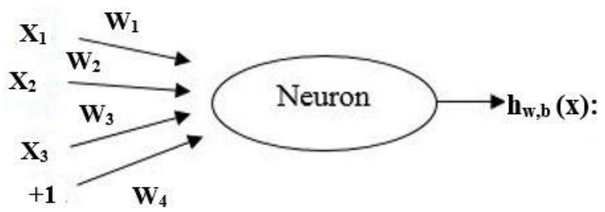


Figure III-2: The simplest possible neural network

There are different types of activation functions. The most common one is the sigmoid function which is defined as below:

$$f(\,z) = \frac{1}{1+\exp(-z)} \tag{3.5}$$

With the sigmoid function the single neuron finds a mapping between the output and inputs defined by logistic regression. The other options for an activation function are the linear, hyperbolic tangent (tanh) and the rectified linear functions. The tanh (z) is a rescaled version of sigmoid function and defined as:

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{3.6}$$

The rectified linear function is partly linear and reaches a saturation at 0 when the input is negative:

$$f(z) = \max(0, x) \tag{3.7}$$

The recent studies show that this function works better for deep neural networks. Figure III-3 shows these activation functions with their output ranges.

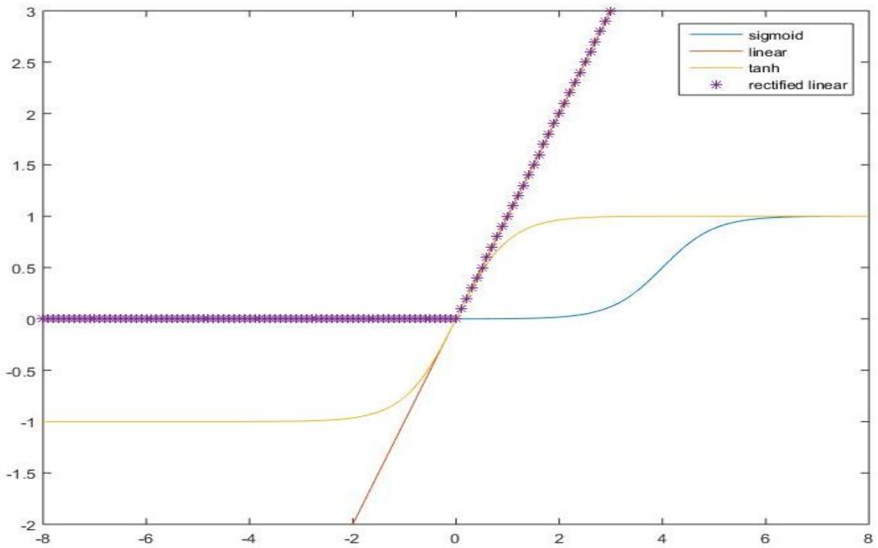


Figure III-3: Different activation functions

It is also very useful for the future calculations to consider the following equations in regard to the derivative of the two mostly used activation functions:

The derivative of a sigmoid function is given by:

$$f'(z) = f(z)(1 - f(z))$$

And the derivative of tanh function is given by:

$$f'(z) = 1 - (f(z))^2$$

A neural network is normally more complicated than the one presented in the figure III-2. Generally, an ANN consists of different layers and weighted connections as the main components. The behavior of the network strongly depends on the relations between these components. In general, there are three types of neuronal layers in ANNs:

- Input layer: which receives the raw input data and feeds it to the network
- Hidden layers: The function of these layers is determined by the inputs and the weights associated to each neuron in the hidden layer and also the relation between them. The weights of the connections between the input layer and hidden layers can determine which hidden layer should be activated or inactivated. This layer is called hidden, because its values are not observed in the training set.
- Output layer: which normally generates the output. The activity of this layer depends on the connection weights between the hidden layers and output neurons.

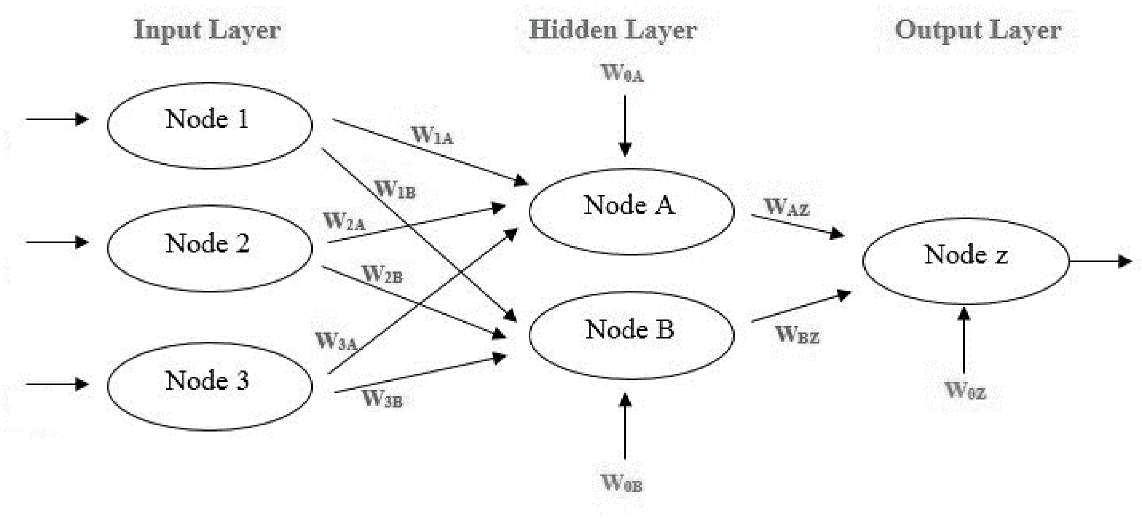


Figure III-4: Structure of a small artificial neural network

Figure III-4 shows how a small ANN is made by these three layers. Let's n_1 be the number of layers in a neural network. In the ANN presented in the above figure n_1 is equal to 3, since we have three layers. Let's label the layers with L_i, so the input layers will be L_1 and the output layer will be Ln_1 and in this case L_3. The ANN has the parameters $W_{i,}$ which denote the weights of the connections between two neurons and *b* is the bias associated to each neuron and will be shown by:

$W^{(l)}ij$ = Weight of the connection between the neuron *i* in the layer *l* and the neuron *j* in the layer *l+1*.

$b^{(l)}i$ = Bias associated with the neuron i in the layer l.

So in the presented ANN, we have $W^{(1)} \in R^{3\times2}$ and $W^{(2)} \in R^{1\times2}$. The activation function or the output of the each neuron in layer l is shown by $a^{(l)}i$ which is equal to x_i in the layer 1. Given a fixed setting for these parameters, it is possible to calculate a hypothetical output $h_{w,b}(x)$ as following:

$a^{(2)}_1 = (W_{AZ}=)\ f(\ W^{(1)}_{1A}\ .\ x_1 + W^{(1)}_{2A}\ .x_2 + W^{(1)}_{3A}\ .x_3 + W_{0A})$

$a^{(2)}_2 = (W_{BZ}=)\ f(\ W^{(1)}_{1B}\ .\ x_1 + W^{(1)}_{2B}\ .x_2 + W^{(1)}_{3B}\ .x_3 + W_{0A})$

$h_{w,b}(x) = a^{(3)}_1 = f(\ W^{(2)}_{AZ}.\ a^{(2)}_1 + W^{(2)}_{BZ}\ .\ a^{(2)}_2 + W_{0Z})$

Assuming that $z^{(l)}_i$ is equal to the total weighted sum of inputs to the neuron i in the layer l (including the bias term), we will have $a^{(l)}i = f\ (z^{(l)}_i)$ with:

$$z_i^l = \sum_{j=1}^{n} W_{ij}^{l-1} x_j + b_i^{l-1}$$

Using this compact notation, the above equations can be written as:

$z^{(2)} = W^{(1)}\ .\ x + b^{(1)}$

$a^{(2)} = f(z^{(2)})$

$z^{(3)} = W^{(2)}\ a^{(2)} + b^{(2)}$

$h_{w,b}(x) = a^{(3)} = f(z^{(3)})$

This form is called ***forward propagation***. Generally, for an ANN with n_1 layers, we can write the equations for the layer l and the activations in layer $l+1$:

$a^{(1)} = x$

$z^{(l+1)} = W^{(l)}\ a^{(l)} + b^{(l)}$

$a^{(l+1)} = f(z^{(l+1)})$

In some cases the architecture of an ANN can be much more complicated and contain several hidden layers which leads to more complicated connection patterns between the neurons. The most common architecture, however, is the same as the one presented before, i.e. an ANN with n_l layers, in which the L_1 is the input layer, the last layer is the output and the layer between them are connected together. In such a structure, it is possible to calculate the output of layer L_2 simply by applying the above equations (forward propagation) on the activities in layer L_1 and then continue to calculate the output of layer L_3 in the same way. This is an example of a ***feedforward*** neural network, since the connections between the units do not form a cycle or loops. In a feedforward ANN, the input is presented to the first layer and the data is propagated forward through the following layers and finally the last layer presents the output. As such, it is different from ***recurrent neural networks*** where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behavior.

It is also worth to note, that the ANNs can have multiple output neurons which normally happens when the prediction of multiple outputs is of interest. (Andrew Ng, 2016) (Negnevitsky, 2011)

3.4. ANNs' Training Process

Neural networks are divided into four main categories based on the training method:

- Fixed weights: In this type, there is no learning process involved and the values of the weights do not update. The weights in this method are usually determined from either the autocorrelation or cross-correlation formulation. The application of this training type is normally in optimization of the information (size reduction, classification and compression) and also in conjunctive memories.
- Unsupervised learning: The weights in this method are corrected and updated based on unlabeled input data. Here, the relation between input and output in unclear and there is no evaluation of output accuracy, since there is no optimal output defined for the system, so that the weights can only be updated based on the patterns and relations identified in the input data. The central application of unsupervised training is to extract the characteristics of the input data based on clustering, classification or similarity recognition (forming groups with similar patterns). However, no output classes will be defined in correspond to the patterns in the input data. The learning in unsupervised training is usually based on the best-match method. The ANN changes its weights based on the output resulted from the inputs, so that in the next similar situation, there is a good response to a similar inputs. This is how the ANN is trained to respond to the inputs. Basically in training, it is important to identify the neuron with the maximum

initial activation based on the dominant neuron techniques. Thus, one of the most important parts in unsupervised trainings is finding the most dominant neuron.

- Supervised learning: This learning process involves pair of input and corresponding output. The weights of the ANN are changed and updated until the difference between the output of the network based on the training patterns and the desired outcomes stays in an acceptable range. The most popular supervised network training is the multilayer back-propagation which will be discussed in the next part. This method is mostly used for the applications which involve a large number of classes and complex classifications. The supervised training method are widely used for pattern and speech recognition tasks.
- Reinforcement learning: This training method is a state between supervised and unsupervised learning in which the quality of the system performance is improved step by step over time. There are no educational patterns, but using a signal called the critique is an expression of the good or bad behavior of the system. (Leondes, 2001) (Negnevitsky, 2011)

3.4.1. Backpropagation Algorithm

One of the most common methods in training an artificial neural network is the so-called ***Backpropagation Algorithm***. This algorithm is basically a method to evaluate the error contribution of each neuron after a batch of input data is analyzed. After computing all the activations through the network including the output value, the difference between the network's activation and the true target value will be measured. If more than one neuron is involved, the error is measured based on a weighted average of the error terms of the nodes that used the previous activation as their input.

Suppose that there is a set of training data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ from m examples with a squared-error cost function $J(W, b; x, y)$ defined as below:

$$J(W, b; x, y) = \frac{1}{2}\left\| h_{W,b}(x) - y \right\|^2 \tag{3.8}$$

Considering all of the *m* data batches, the overall cost function will be defined as:

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^{(i)},y^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2 \tag{3.9}$$
$$= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2$$

The first term in the above formula is an average sum of the squared-error term and the second term is the so-called "***weight decay term***", which is regularization term to prevent overfitting in the training process and controls the relative prominence of the two terms.

The presented function, which is also known as cost function, is normally used for classification and regression. Based on the type of activation function, the value of *y* can represent the class labels, e.g. *y= -1* and *y= +1* for *tanh* activation function. For the regression problems with the same activation function, the output will be scaled to stay between [-1, 1] range.

The goal is to minimize the cost function and hence train the neural network with the best possible weights. This optimization will be done by help of gradient descent algorithm. For this algorithms, backpropagation algorithm is needed to calculate the partial derivatives. Generally the following steps will be performed in backpropagation algorithm:

1. Perform a feedforward signal pass, and calculating the activations for layers L_2, L_3, and so on up to the output layer L_{nl}.
2. For each output neuron *i* in the layer n_l set the error term $\delta_i^{n_l}$ as follows:

   $$\delta_i^{n_l} = \frac{\partial}{\partial {z_i}^{(n_l)}}\frac{1}{2}\left\|y - h_{W,b}(x)\right\|^2 = -\left(y_i - a_i^{(n_l)}\right).f'\left({z_i}^{(n_l)}\right)$$

   for $l = n_l-1, n_l-2, n_l-3, \dots, 2$

3. For each neuron *i* in the layer *l* set

   $$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)}\,\delta_j^{(l+1)}\right).f'\left({z_i}^{(l)}\right)$$

4. Finally calculate the desired partial derivatives, which are given for instance as:

$$\frac{\partial}{\partial W_{ij}{}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)}$$

$$\frac{\partial}{\partial b_i{}^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)}$$

Once these partial derivatives are computed, the iteration of gradient descent optimization algorithm can be calculated to reach the best weights for training the ANN. (Andrew Ng, 2016) This algorithm was used for the training process of the system using the prepared datasets in this chapter.

3.4.2. Perceptron Model

A single layer perceptron artificial neural network is actually a type of linear classifier. Perceptron is a unit which includes an algorithm for a supervised learning. The computational unit in a Perceptron ANN takes the real values of the inputs and calculates a linear combination among them. This unit is normally a simple neuron layer that classifies inputs into one or two categories. It uses a step activation function. Each external input is weighted with an appropriate weight and the sum of the weighted inputs is sent to the hard-limit transfer function which also has an input of 1 transmitted to it through the bias. If the sum of the weighted inputs (X) is greater than the threshold of b, then it returns 1; otherwise it returns 0:

$$X = \sum_{i=!}^{m} X_i W_i \tag{3.10}$$

$$Y = \begin{cases} 1, & x > b \\ 0, & x \leq b \end{cases}$$

Perceptron learning process:

1. The threshold value will be set and random values will be assigned to the weights of inputs. Weights may be initialized to 0 or to a small random value.
2. A sample of the training data will be given to the perceptron and the output of the classifier will be observed.
3. If the output is correct, the learning process stops.
4. Otherwise, the weights will be changed to reach the correct classification based in this input dataset.

The modification of the weights is done based on the following formula:

$$W_i(n) = (\beta . x_i . e) + W_i(n-1) \tag{3.11}$$

Where:

β = learning rate (normally equal to 0.1)

x_i = Input *i*

$e = d - y$; d goal output and *y* output from the process

$W_i(n)$ = the weight of the input *i* in the iteration *n*

The single perceptron ANN is very practical for linear classifications. However, in most problems the linear classification is not enough and the ANN should be trained better to achieve more accurate results. Here, the Multi-Layer Perceptron (MLP), which is a recurrent neural network, is a good answer to solve such non-linear problems. These networks are able to make a nonlinear map with acceptable accuracy by choosing the appropriate number of layers and neuros, which are often not too high.

Figure III- 5 shows the structure of a MLP.

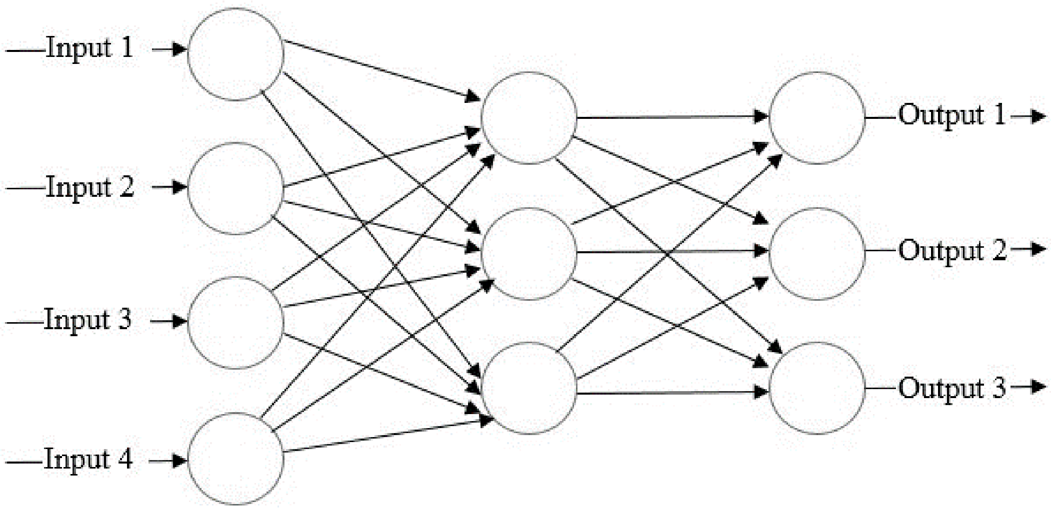


Figure III-5: Structure of a Multi-Level Perceptron

Generally a recurrent neural network has the following features:

- The activation function of the neurons is normally sigmoid or tanh.
- The network has one or several hidden layers of neurons, which are neither inputs nor outputs. These layers enable the network to learn complex non-linear functions
- The sample data are fed forward through the input node to the output layers.

- There will be connections from a previous layer to all of the neurons in the next layer. Normally, the recurrent ANNs have one input layer, one or two hidden layer and one output layer. Each of these layers contain 10 to 1000 neurons depending on the application and complexity of the system.

The MLP networks work almost the same as the single perceptron neural networks. The only difference is that in MLP each neuron has more connections and as a result more weights which will be modified in the training process. Generally, the learning algorithm in these networks is also back propagation which follows these steps:

1. All the weight will be chosen randomly, same as the process done in single perceptron network.
2. Training datasets are fed to the input layers and the signal will be through to the output layer.
3. The difference between the computed output and the goal value (errors) will be sent back to the input layers.
4. The new values for the weights will be modified based on step 3.
5. This algorithm continues till it reaches one of the ending conditions, which can be time, a specific error rate or the number of iterations. (Negnevitsky, 2011) (Gang Li, 2007) (Andrew Ng, 2016)

This structure is used in this chapter to train the system and test the results.

4. Results of Load Predictions Based on ANNs

As mentioned in the last part, the sample data from the electrical load of CUTEC institute in 2014 was used as training sets for a MLP neural network with back propagation algorithm. The MLP ANN was fed systematically with the categorized data, which has been done based on the lags achieved by the mutual information function. As explained in part 2.2, these lags are different for each time horizon and the network will be fed differently for each type. Figure III-6 to III-8 show the mutual information plots for different time horizon. Based on these plots, the numbers of the best lags and their values were determined by a trial and error process.

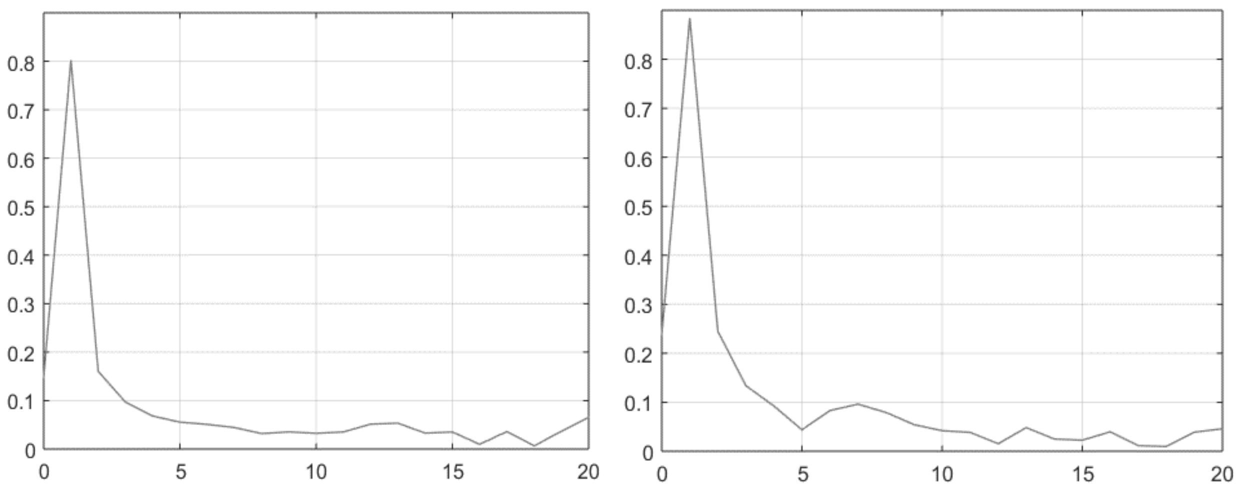


Figure III-6: Time-lagged plot of mutual information function for two different samples of weekly data

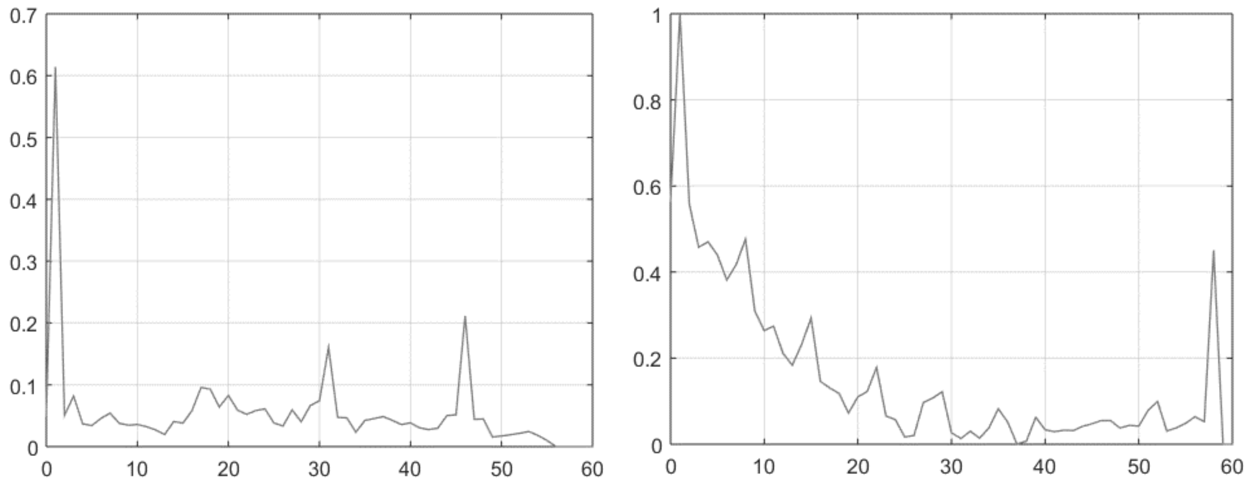


Figure III-7: Time-lagged plot of mutual information function for two different samples of monthly data

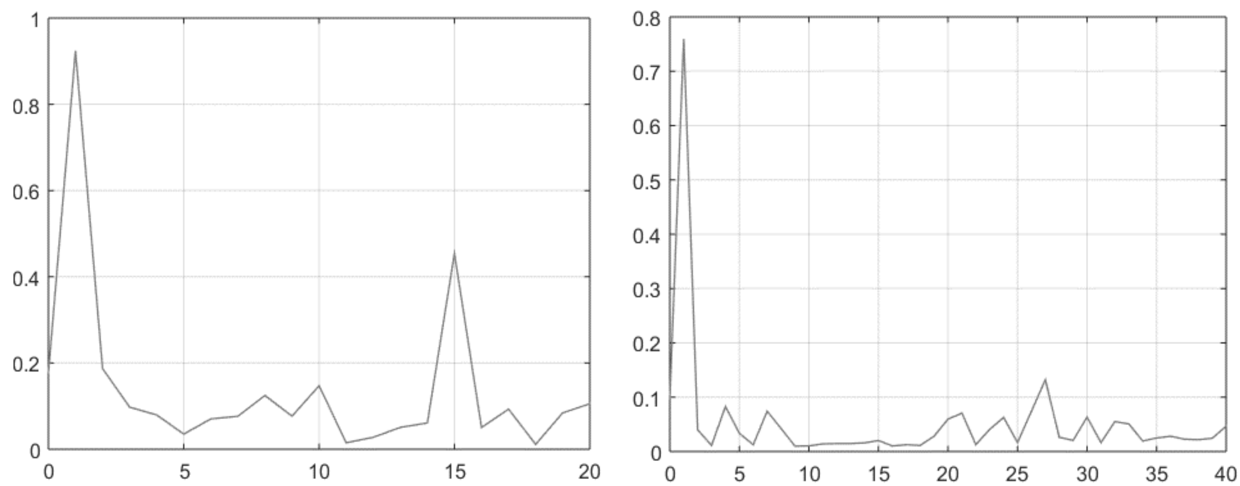


Figure III-8: Time-lagged plot of mutual information function for two different samples of seasonal data

The final optimal lags for weekly, monthly and seasonal load prediction are reviewed in the table III-4:

Table III-4: Optimal Lags calculated by mutual information function

Type of Prediction	Time series classification Lags (delays)
Weekly	[1 1]
Monthly	[1 3 8]
Seasonal	[1 8 10]

The MLP network has then been trained with the classified data. To evaluate the results, 75 percent of the whole data was chosen randomly to train the network and the output of the model for the rest 25 percent has been compared with the actual recorded load and the performance is assessed by the following parameters:

- Mean Error (ME)
- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Standard Deviation Error (STD Error)
- Mean of Absolute Percentage Error (MAPE)

4.1. Weekly Prediction Results:

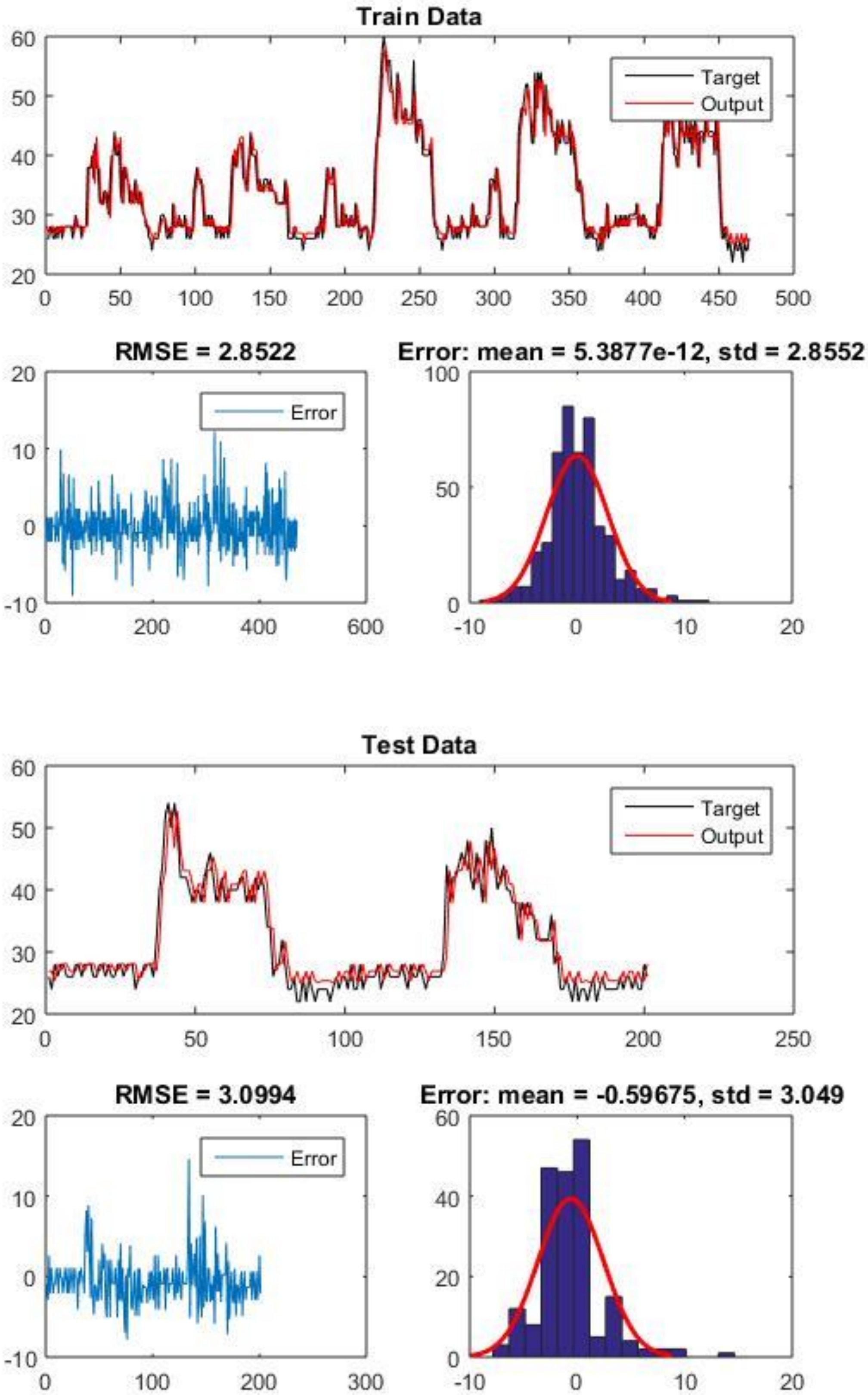
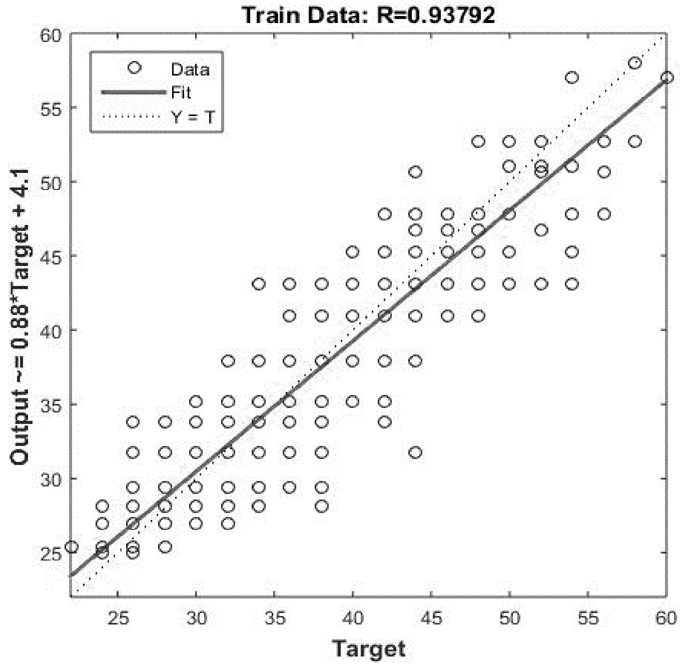


Figure III-9: Target –Output diagram for train and test data in weekly load prediction



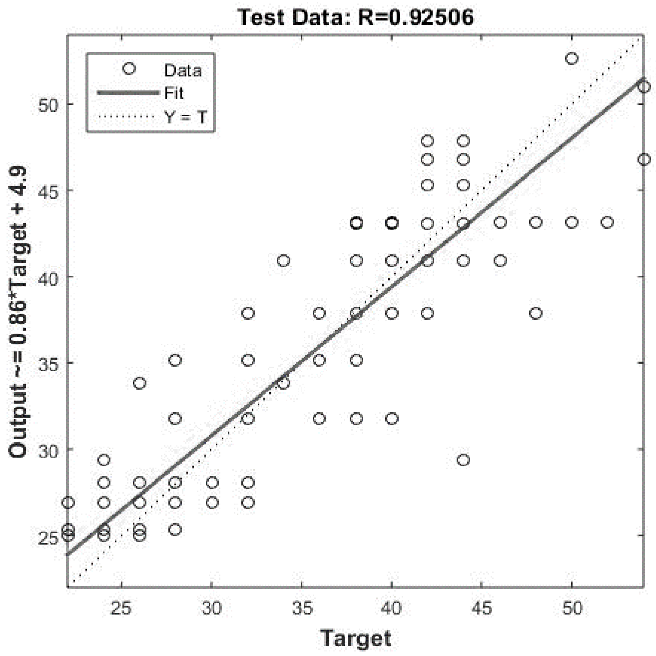


Figure III-10: Regression diagram for train and test data in weekly load prediction

Table III-5: Weekly prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	-5.387e-12	8.1348	2.8522	2.8552	2.0879
Test	0.5968	9.6063	3.0994	3.0490	2.3288

4.2. Monthly Prediction Results:

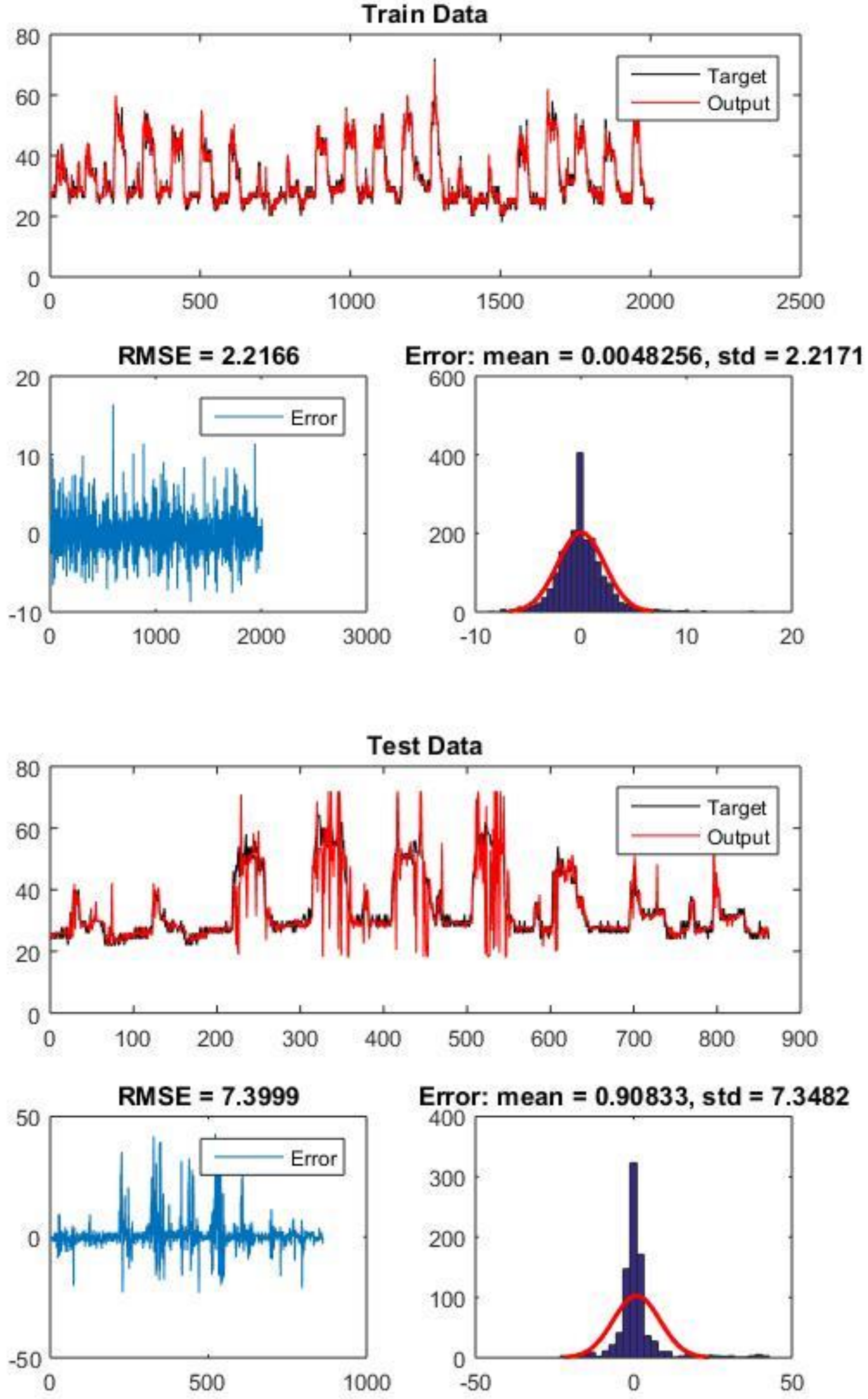
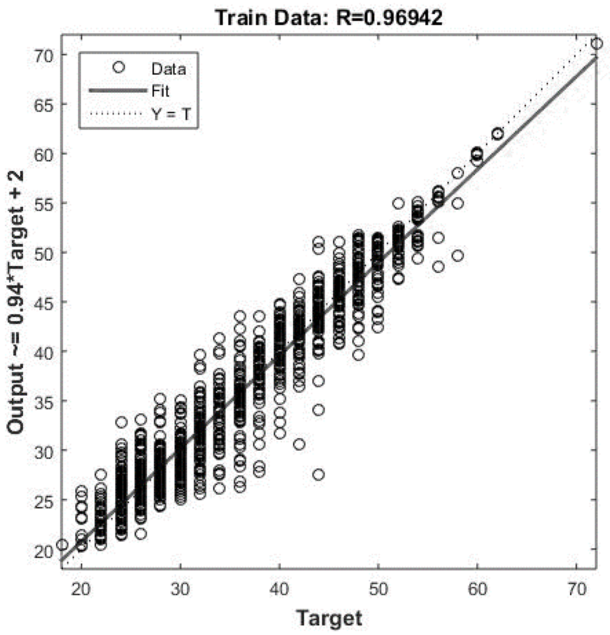


Figure III-11: Target –Output diagram for train and test data in monthly load prediction



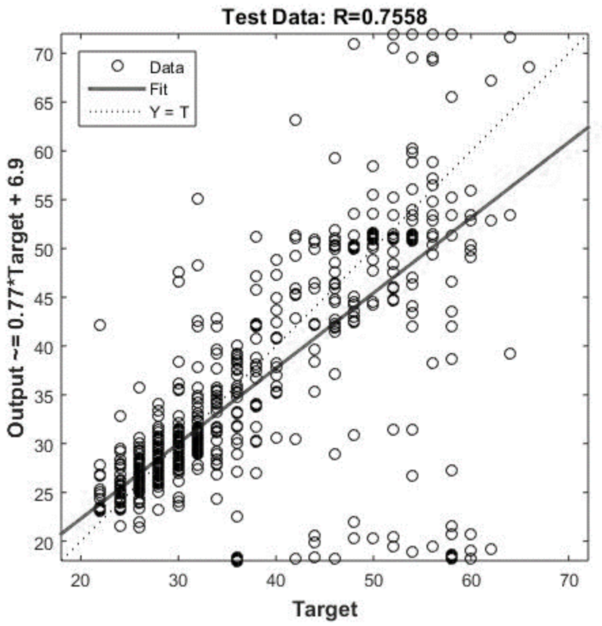


Figure III-12: Regression diagram for train and test data in monthly load prediction

Table III-6: Monthly prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	-0.0048	4.9131	2.2166	2.2171	1.5581
Test	-0.9083	54.758	7.3999	7.3482	3.7342

4.3. Seasonal Prediction Results:

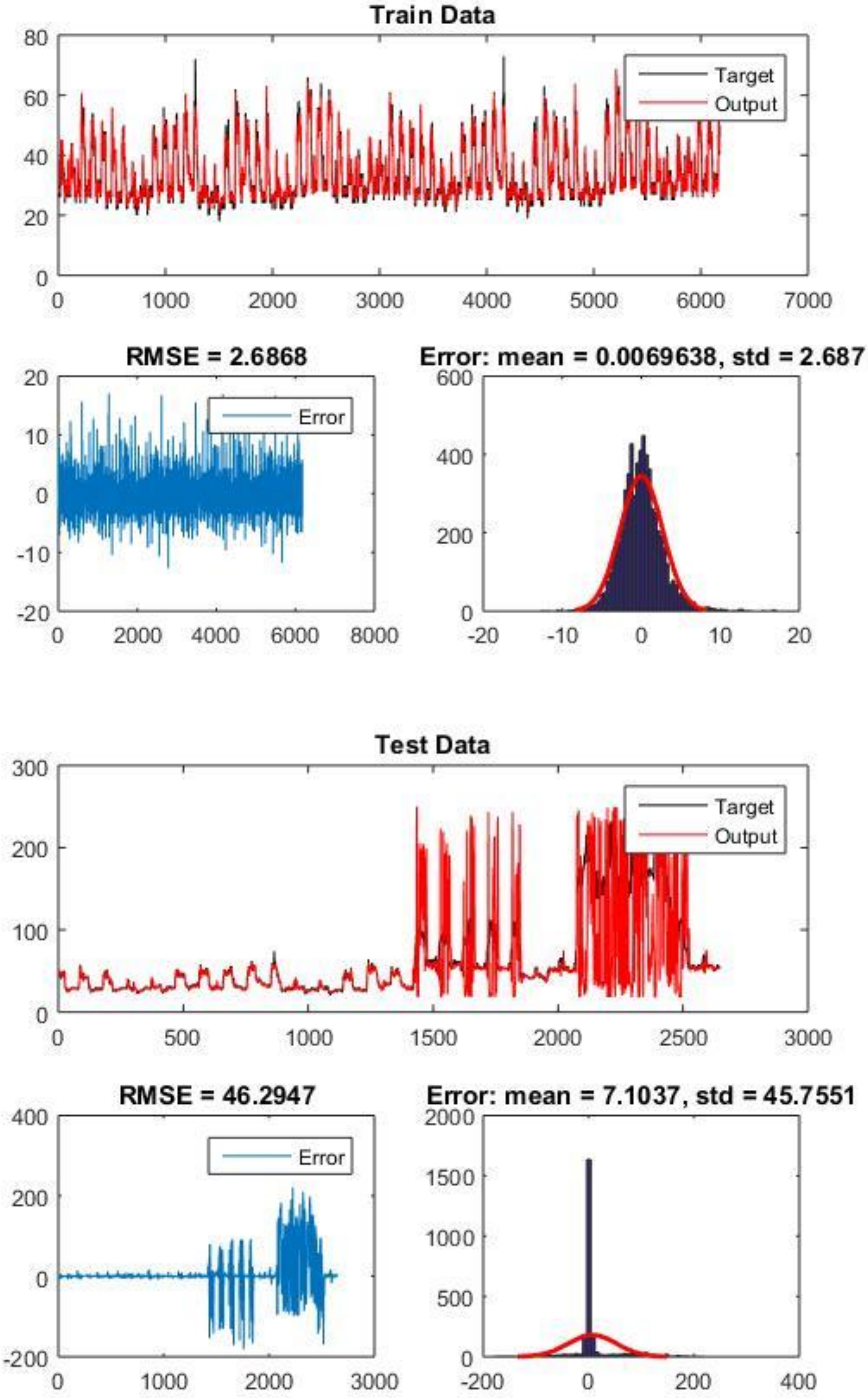


Figure III-13: Target –Output diagram for train and test data in seasonal load prediction

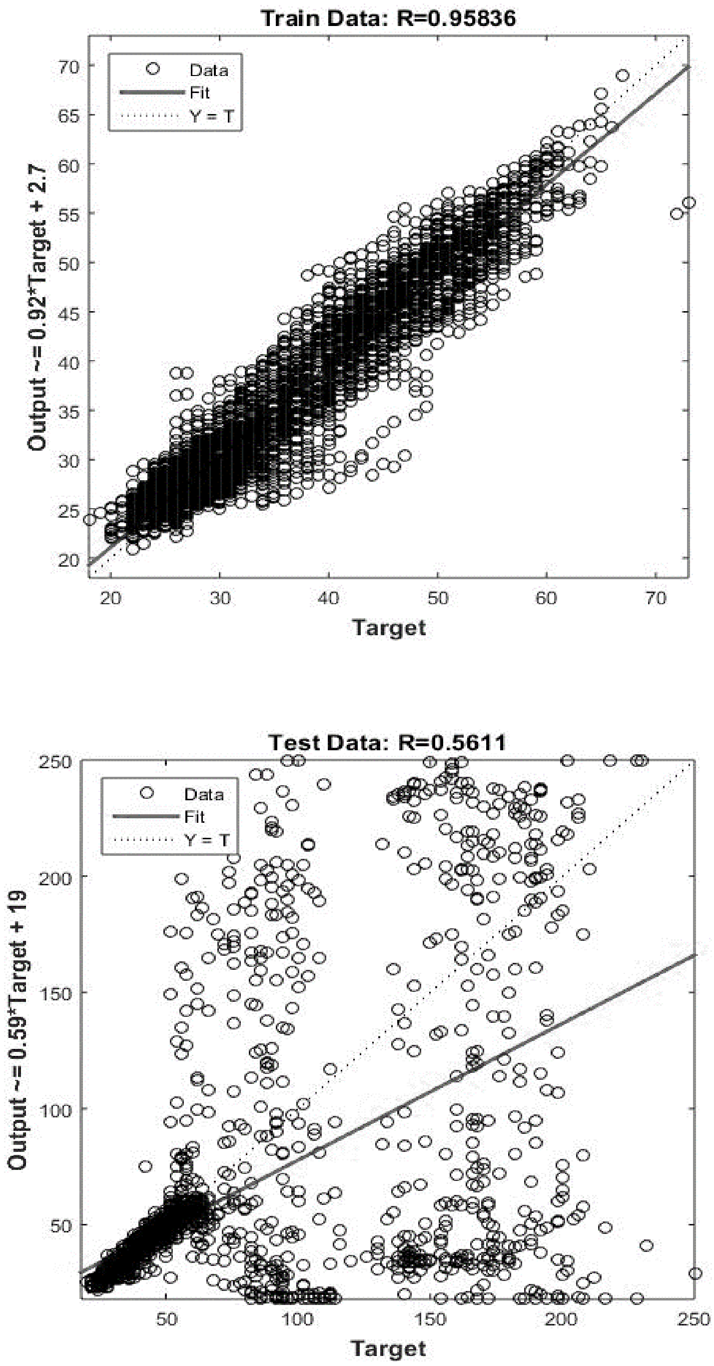


Figure III-14: Regression diagram for train and test data in seasonal load prediction

Table III-7: Seasonal prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	-0.0070	7.2191	2.6868	2.6870	1.9940
Test	-7.1037	2.14 e+03	46.294	45.755	21.969

The presented results proves that the categorization of the input based on the similarities and systematic feeding of them into an ANN system makes significant difference in prediction accuracy in comparison to the presented results from the other studies using the same approach. However, the ANN has some disadvantages which can affect the accuracy of training. Overcoming these problems may lead to a better prediction model which is the intention of this dissertation.

5. Neural Networks Drawbacks

Despite the advantages in comparison to classic methods, there are some drawbacks in methods based on the artificial neural networks which may affect the training process and as a result, the accuracy of the prediction may decrease. Some of these disadvantages are listed below:

- The ANNs have a black-box nature, i.e. it is not possible to completely control the way the neurons are connected in the network.
- There are no specific rules or guidelines for an optimal network design for different applications.
- The ANN approach alone is normally not enough for the modeling problems, especially in physical modeling.
- The accuracy of the results depend largely on the size of the training data.
- The training of the network in some specific problems may be very hard or even impossible.

The researchers have tried to overcome these disadvantages by introducing new methods or combining the ANN-based methods with other approaches. (Kim, Youn, & Kang, 2000) In the next chapter, one of such approaches will be introduced and the results of the load forecasting based on that method with the same dataset will be compared to the ones presented in the part 4.5 in this chapter.

IV. Chapter 4

1. Adaptive Neuro-Fuzzy Inference System (ANFIS)

As discussed in chapter III, the model obtained with neural network has brilliant self-learning and adaptive ability but it is not understandable in terms of physical parameters (black box model). On the other hand, Fuzzy Logic is considered as an effective tool in modeling complex systems, in which problems are difficult to understand or dependent on reasoning, decision making and human inference. The fuzzy-based modeling consists of some if-then rules, which reflect the reasoning ability and are almost natural language, but it lacks the adaptive ability and cannot learn the rules by itself. In recent years, there have been many efforts to achieve better accuracy in modeling by combining the common methods in load forecasting and artificial intelligence. One of the best ideas was the fusion of neural networks and fuzzy logic in ***Neuro-Fuzzy*** models. Neural network and fuzzy logic systems are the universal approximators with capability of identifying non-linear relationship between inputs and targets. (Honade & J., 2015) (A Khosravi, 2010) Adaptive Neuro-Fuzzy Inference System (ANFIS) was proposed by Jang Roger in 1993, which overcame the problems of neural networks and fuzzy logic by combining the self-learning ability of ANN and the logical inference ability and also robustness and ease in implementing the rules bases of fuzzy systems. (unkas, 2015) (Pandian SC, 2006) The ANFIS systems have high efficiency and great convenience in facing problems of non-linearity and uncertainty in data. (Wang XiangJun, 2012)

The fuzzy part of the ANFIS contains input and output variables, membership functions, fuzzy rules and inference method. The training inputs will be fed to the ANFIS and they can affect the outputs. In most of the ANFIS-based load forecasting methods, the training data are the climate change, the day of the week, etc. Here, the electrical load data will be analyzed by itself with no other parameters involved and the training process in ANFIS will be done by the load time series alone. This is a new approach in using ANFIS for the load forecasting and the classification of the data and the systematic feeding of data to ANFIS is the novel idea behind this method. Part 1.1 explains the structure of ANFIS in detail.

1.1. ANFIS Structure

As mentioned before, ANFIS is an adaptable and trainable network with a similar functionality to a fuzzy inference system. Generally, ANFIS performs the following steps:

- Receiving the training data
- Mapping the input characteristics to membership functions
- Applying specific rules on the input data from the previous step
- Define outputs based on the rules
- Mapping the output characteristics to output membership functions
- Calculating the total output of the whole network (in form of a single valued output)

The fuzzy rules inside the ANFIS define the output based on a specific value of the inputs and have a form of if-then rule:

$$Rule\ 1: if\ (x\ is\ A_1)\ and\ (y\ is\ B_1)\ then\ f_1 = p_1x + q_1y + r_1 \tag{4.1}$$
$$Rule\ 2: if\ (x\ is\ A_2)\ and\ (y\ is\ B_2)\ then\ f_2 = p_2x + q_2y + r_2$$

In the above equations, *x* and *y* are inputs, *A* and *B* are fuzzy sets, f_i is output from fuzzy region specified by fuzzy rules, p_i, q_i and r_i are the modifying factors which are determined during the training process. (Bakirtzis A. G. et al., 1995) (Seema Pal, 2015)The structure of an ANFIS system with these two rules is shown in the figure IV-1:

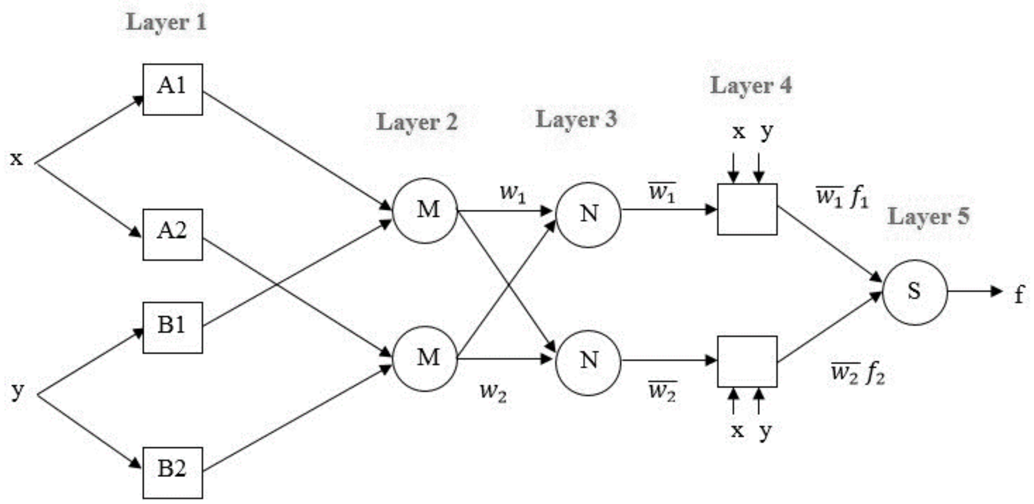


Figure IV-1: The structure of ANFIS (circles represent the fixed nodes and squares represent the adaptive nodes) (Junran Peng, 2017)

The first layer of the ANFIS or ***Fuzzification*** is the membership function layer of the input parameters. All the nodes in this layer are adaptive nodes. There are various membership functions, such as the triangular, trapezoidal, pi shaped, z shaped, s shaped, generalized bell function, etc. in the context of this dissertation, the Gaussian membership function (Equation (4.3)) has been adopted[4] which results the outputs of the first layer to be in the following form:

$$O_i^1 = \mu A_i(x) \qquad i = 1,2 \tag{4.2}$$
$$O_i^1 = \mu B_i(y) \qquad i = 1,2$$

Where:

$$\mu = \exp\left(-\left\{\left(\frac{x-c_i}{a_i}\right)^2\right\}^{b_i}\right) \tag{4.3}$$

In this equation, a_i, b_i and c_i are the so-called “front parameters”, which control this function. The parameters in this layer are called the “premised parameters”. There are two main steps in the processes based on the ANFIS model. The first step is training, in which the membership function parameters are modified, so that the desired pattern and relation between the inputs and outputs is learned by the system. The training data batches will be presented to the network many times (iterations) until the desired results are obtained (In most cases it is the mean square error between output and target). The next step is testing the trained network with the rest of the data.

The output of the second layer is the multiplication of input signals which is actually equivalent to the "if" part of the rules. In this layer, the weights of each rule will be computed through a fuzzy AND operation. The nodes in this layer are all fixed nodes. The following equation presents the output of this layer:

$$O_i^2 = w_i = \mu A_i(x)\, \mu B_i(y) \qquad i = 1,2 \tag{4.4}$$

The third layer normalizes the values from the previous layer. The output of this layer can be shown by:

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1+w_2} \qquad i = 1,2 \tag{4.5}$$

[4] This function is chosen over the generalized bell-shaped function by means of intensive trial and error procedure.

The ***Deffuzification*** process takes place in the fourth layer. All the nodes in this layer are also adaptive nodes and the corresponding formula for the output from this layer can be expressed as:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \quad i = 1,2 \tag{4.6}$$

Where $\overline{w_i}$ is the output from the third layer and p_i, q_i and r_i are the controlling factors of this node or the so-called "later parameters".

The final layer presents the overall output of the whole network and it contains a single fixed node which sums up the results from the previous layer:

$$O_i^5 = \sum_i \overline{w_i} \ f_i = \frac{\sum w_i f_i}{\sum w_i} \qquad i = 1,2 \tag{4.7}$$

This network is based on a particular type of fuzzy systems with Takagi-Sugeno rules. (Amit Jain, 2009) (Thai Nguyen, 2014) (Esa Aleksi Paaso, 2013)

Figure IV-2 shows an example of ANFIS model with 2 input variables. In this example, each input has three membership functions, so that a total 9 rules should be defined:

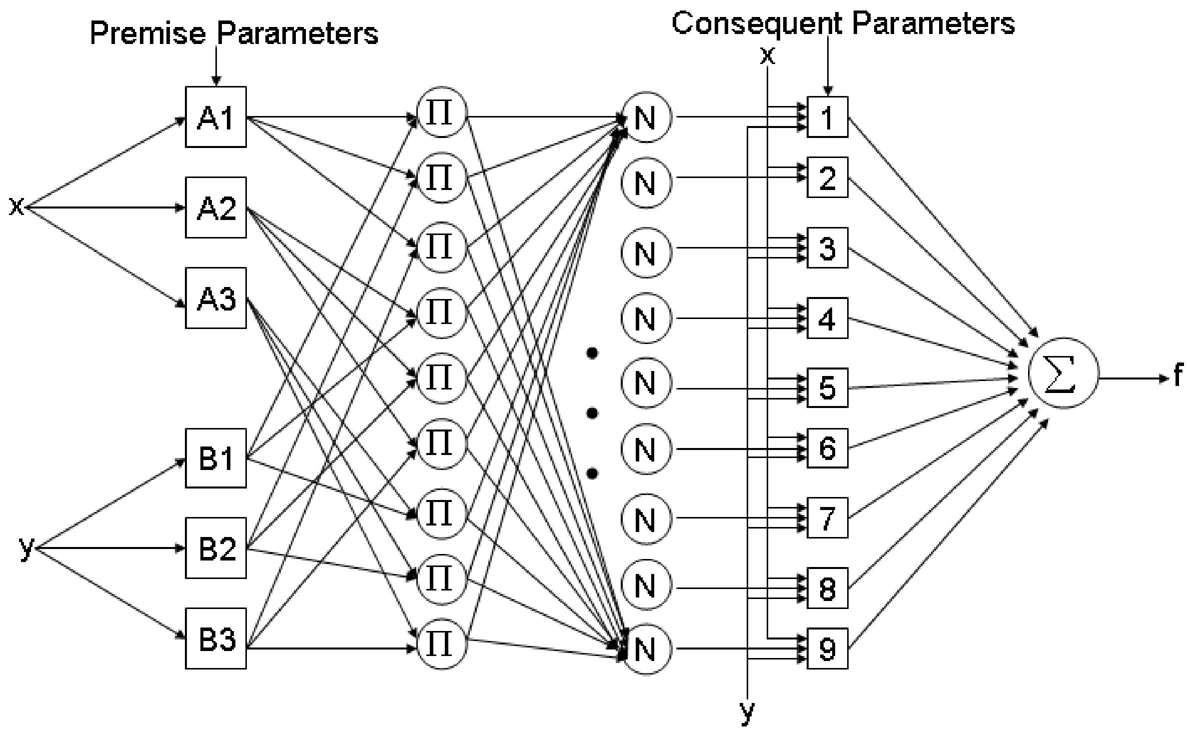


Figure IV-2: ANFIS network structure with 2 input variables (Thai Nguyen, 2014)

1.2. ANFIS Training Methods

The most important role of the ANFIS network presented in the last part is the fuzzy inference system (FIS) which tries to learn the algorithm or any possible relation between input/output and adjust the front and later parameters so that the final outputs of the ANFIS match the training aims. Assuming that the membership function of the front parameters in the second layer is stable, the final output of the ANFIS network can be given by:

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \tag{4.8}$$

Considering the equation (4.5), the above formula can be written as:

$$f = \overline{w_1} f_1 + \overline{w_2} f_2 \tag{4.9}$$

And replacing f_1 and f_2 with the fuzzy rules introduced at the beginning of the chapter, we will have:

$$f = \overline{w_1}(p_1 x + q_1 y + r_1) + \overline{w_2}(p_2 x + q_2 y + r_2) \tag{4.10}$$

The rearranged form of the above equation can be written as:

$$f = (\overline{w_1} p_1)x + (\overline{w_1} q_1)y + (\overline{w_1} r_1) + (\overline{w_2} p_2)x + (\overline{w_2} q_2)y + (\overline{w_1} r_2) \tag{4.11}$$

Here, the p_1, q_1, r_1, p_2, q_2 and r_2 form a modifiable linear parameter set. This function can now be optimized by applying various optimization methods. However, if the front parameters are not stable, the optimization process will be more complicated. In the next part, two of the most common methods in optimization of ANFIS will be introduced and finally the combination of these two method will be discussed as a better solution which is also used in the course of the model developed for this dissertation.

1.3. Least Square Error Method

The Least Square Error method (LSE) is a common algorithm in regression analysis specially to find the solutions in the systems where the number of equation are more than the unknowns. In this method the overall solution minimizes the sum of the squares of the errors from every single equation.

In general, the output of a linear model is given by the following formula:

$$y = \theta_1 f_1(u) + \theta_2 f_2(u) + \cdots + \theta_n f_n(u) \tag{4.12}$$

Where u is the input vector, f_i is a known function for training and θ_i is the unknown parameter, which has to be estimated. To identify these parameters, some training data in form of $(u_i; y_i) \quad i = 1, \dots, m$ is needed. Replacing the input/output datasets in the main formula, the following linear formulas will be achieved:

$$\begin{cases} f_1(u_1)\theta_1 + f_2(u_1)\theta_2 + \cdots + f_n(u_1)\theta_n = y_1 \\ f_1(u_2)\theta_1 + f_2(u_2)\theta_2 + \cdots + f_n(u_2)\theta_n = y_2 \\ \vdots \\ f_1(u_m)\theta_1 + f_2(u_m)\theta_2 + \cdots + f_n(u_m)\theta_n = y_m \end{cases} \tag{4.13}$$

Rewriting the above formulas in matrix form, we will have:

$$\boldsymbol{A\theta} = \boldsymbol{Y} \tag{4.14}$$

Where,

$$\boldsymbol{A} = \begin{bmatrix} f_1(u_1) & \cdots & f_n(u_1) \\ \vdots & \ddots & \vdots \\ f_1(u_m) & \cdots & f_n(u_m) \end{bmatrix}$$

And

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \ \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Based on these equations, the matrix $\boldsymbol{\theta}$ can be calculated as:

$$\boldsymbol{\theta} = \boldsymbol{A}^{-1}\boldsymbol{Y} \tag{4.15}$$

As mentioned before, the input/output datasets are normally more than the number of linear equations, so that finding $\boldsymbol{\theta}$ is possible. However, sometimes the data may contain noise or the model may not be able to accurately determine the output. For such cases the above equation will be modified to:

$$\boldsymbol{A\theta} + \boldsymbol{e} = \boldsymbol{Y} \tag{4.16}$$

Where $\boldsymbol{e}$ denotes the possible error or noise inside the data.

To find the optimal matrix of parameters, we look for $= \widehat{\boldsymbol{\theta}}$, which minimizes the sum of the squares of the errors:

$$E(\boldsymbol{\theta}) = \sum_{i=1}^{m}(y_i - a_i^T\theta)^2 = e^Te = (\boldsymbol{Y} - \boldsymbol{A\theta})^T(\boldsymbol{Y} - \boldsymbol{A\theta}) \tag{4.17}$$

And if the matrix $\boldsymbol{A^T A}$ is invertible, then:

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{A^T A})^{-1} \boldsymbol{A^T Y} \tag{4.18}$$

Another similar method to calculate the parameters, which is specifically applicable when the matrix $\boldsymbol{A^T A}$ A is not invertible, is the so-called Recursive Least Square Error (RLSE). This method has a higher convergence speed. Furthermore, it is more convenient for the systems in which the data arrives sequentially and the model is being updated by new sets of incoming data. Applying the RLSE method, there is no need to restart from scratch the model estimation and calculating all the parameters, but simply to update the model on the basis of the newly collected data.

The recursive formulas for this method are given below:

$$S_{k+1} = S_k - \frac{S_k a_{k+1} a_{k+1}^T S_k}{1 + a_{k+1}^T S_k a_{k+1}} \tag{4.19}$$

$$\theta_{k+1} = \theta_k + S_{k+1} a_{k+1} \left(y_{k+1} - a_{k+1}^T \theta_k \right) \quad for\ k = 0,1, \dots , m-1$$

The initial conditions will be set to $\theta_k = 0$ and $S_0 = \gamma\, I$ where I is an identity matrix with m×m dimension and γ is a large integer.

1.4. Gradient Descent Method

Gradient descent is a first order iterative optimization method. Recalling that the membership functions in the ANFIS network are assumed to be Gaussian:

$$\mu_{A_i(x)} = \exp\left(-\left\{ \left(\frac{x - c_i}{a_i} \right)^2 \right\}^{b_i} \right)$$

The efficiency criterion will be defined as the sum of the square errors:

$$E_k = \frac{1}{2} (y_k - O_k)^2 \tag{4.20}$$

Where y_k is the output of ANFIS and O_k is the aimed output.

The chain derivative for an input variable such as B_i in the network presented in figure IV-2 will be given by:

$$\frac{\partial E_k}{\partial B_i} = \frac{\partial E_k}{\partial O_k} \frac{\partial O_k}{\partial B_i} = -\left(y_k - O_k \right) \frac{\partial O_k}{\partial B_i} \tag{4.21}$$

And for the consequent parameters:

$$\frac{\partial E}{\partial B_c} = \frac{\partial E}{\partial O_5}\frac{\partial O_5}{\partial O_4}\frac{\partial O_4}{\partial B_c} \tag{4.22}$$

And for the premise parameters:

$$\frac{\partial E}{\partial B_p} = \frac{\partial E}{\partial O_5}\frac{\partial O_5}{\partial O_4}\frac{\partial O_4}{\partial O_3}\frac{\partial O_3}{\partial O_2}\frac{\partial O_2}{\partial O_1}\frac{\partial O_1}{\partial B_p} \tag{4.23}$$

Considering the equations (4.4) to (4.7), it is possible to compute the relative derivative of layers:

$$\frac{\partial O_5}{\partial O_4} = \frac{\partial(\sum_j f_j \overline{w_J}\ )}{\partial(f_j \overline{w_J})} = 1$$

$$\frac{\partial O_4}{\partial O_3} = \frac{\partial(f_j \overline{w_J})}{\partial(\overline{w_J})} = f_j$$

The derivative of the third layer to the second layer is different for conforming and non-conforming nodes. In the first case:

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial}{\partial w_i}\left(\frac{w_i}{\sum_{j=1}^{n} w_j}\right) = \frac{\sum_{j=1}^{n} w_j - w_i}{\left(\sum_{j=1}^{n} w_j\right)^2}$$

And for non-conforming nodes:

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial}{\partial w_i}\left(\frac{w_i}{\sum_{j=1}^{n} w_j}\right) = \frac{-w_i}{\left(\sum_{j=1}^{n} w_j\right)^2}$$

And finally the derivative of the second layer to the first layer will be:

$$\frac{\partial O_2}{\partial O_1} = \frac{\partial}{\partial A_m}\left(\prod_{A_j \in R(A_m)} A_j\right) = \prod_{A_j \in R(A_m), A_j \neq A_m} A_j$$

Where $A_j \in R(A_m)$ denotes the fuzzy sets which form the fuzzy rules in the ANFIS network.

Consequently, the derivative of sum of the square errors relative to consequent parameters will be defined as:

$$\frac{\partial E}{\partial B_c} = -(d - O)\frac{\partial O_4}{\partial B_c} \tag{4.24}$$

And the derivative of sum of the square errors relative to premise parameters will be:

For conforming nodes:

$$\frac{\partial E}{\partial B_p} = -(d-O) f_i \frac{\sum_{j=1}^{n} w_j - w_i}{\left(\sum_{j=1}^{n} w_j\right)^2} \prod_{A_j \in R(A_m), A_j \neq A_m} A_j \frac{\partial O_1}{\partial B_p} \tag{4.25}$$

And for non-conforming nodes

$$\frac{\partial E}{\partial B_p} = -(d-O) f_i \frac{-w_i}{\left(\sum_{j=1}^{n} w_j\right)^2} \prod_{A_j \in R(A_m), A_j \neq A_m} A_j \frac{\partial O_1}{\partial B_p} \tag{4.26}$$

The derivative of the fourth layer to each one of the consequent parameters $\frac{\partial O_4}{\partial B_c}$ in equation (4.24) will be given by:

$$\frac{\partial O_4}{\partial p_i} = \frac{\partial}{\partial p_i}(f_i \overline{w_i}) = \frac{\partial}{\partial p_i}\left(\overline{w_i}(p_i x + q_i y + r_i)\right) = \overline{w_i} x \tag{4.27}$$

$$\frac{\partial O_4}{\partial q_i} = \frac{\partial}{\partial q_i}(f_i \overline{w_i}) = \frac{\partial}{\partial q_i}\left(\overline{w_i}(p_i x + q_i y + r_i)\right) = \overline{w_i} y$$

$$\frac{\partial O_4}{\partial r_i} = \frac{\partial}{\partial r_i}(f_i \overline{w_i}) = \frac{\partial}{\partial r_i}\left(\overline{w_i}(p_i x + q_i y + r_i)\right) = \overline{w_i}$$

And the derivative of the first layer to each one of the premise parameters $\frac{\partial O_1}{\partial B_p}$ in equation (4.26) will be given by:

$$\frac{\partial O_1}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}}\left(exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)\right) = 2\frac{\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}} b_{ij} exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)}{a_{ij}} \tag{4.28}$$

$$\frac{\partial O_1}{\partial b_{ij}} = \frac{\partial}{\partial b_{ij}}\left(exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)\right) = 2\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}} ln\left(\frac{x-c_{ij}}{a_{ij}}\right) exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)$$

$$\frac{\partial O_1}{\partial c_{ij}} = \frac{\partial}{\partial c_{ij}}\left(exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)\right) = 2\frac{\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}} b_{ij} exp\left(-\left(\frac{x-c_{ij}}{a_{ij}}\right)^{2b_{ij}}\right)}{x-c_{ij}}$$

1.5. Hybrid Method

It was mentioned in the beginning of part 2 that when the front parameters are not stable, the convergence speed will be slower, since the search area is larger. A hybrid algorithm of least square method and gradient descent, which were discussed in part 2.1 and 2.2, have shown great potential to overcome this problem. This algorithm performs a forward delivery process and then a back propagation process, i.e. when the premise parameters are determined, the least square method will be applied to optimize the consequent parameters. After calculating the best consequent parameters, the back propagation process will start and the gradient descent process will adjust the premise parameters based on the fuzzy sets in entry field. In other words, the ANFIS network calculates the consequent parameters in the feed forward process and the premise parameters will be modified by back propagation algorithm according to the output error. (Suparta, 2016) This hybrid approach has shown the best results and highest efficiency in training the neuro-fuzzy networks and it has been applied in the model developed for the load forecasting in this dissertation.

2.Results of Load Predictions Based on ANFIS

The key to an accurate load forecasting based on the method proposed in this dissertation is the systematic feeding of the input datasets to the ANFIS network. As discussed in chapter II and III, the historical load time series of the case study of this dissertation has been analyzed based on concept drifts and the appropriate lags for each type of prediction (weekly, monthly, seasonal and yearly) has been calculated according to the mutual information function, which were presented in part 4 of chapter III. Based on these lags different datasets have been created in which each input is corresponded to two or more relevant outputs, so that the training can be more accurate.

The next parts presents the diagrams and the results of ANFIS network for all the three types of load prediction based on the proposed model. The detailed implementation and MATLAB codes of all the steps including the mutual information and chaos detection (except the ANFIS training and testing process) are available in the appendix A.

2.1. Weekly Prediction Results:

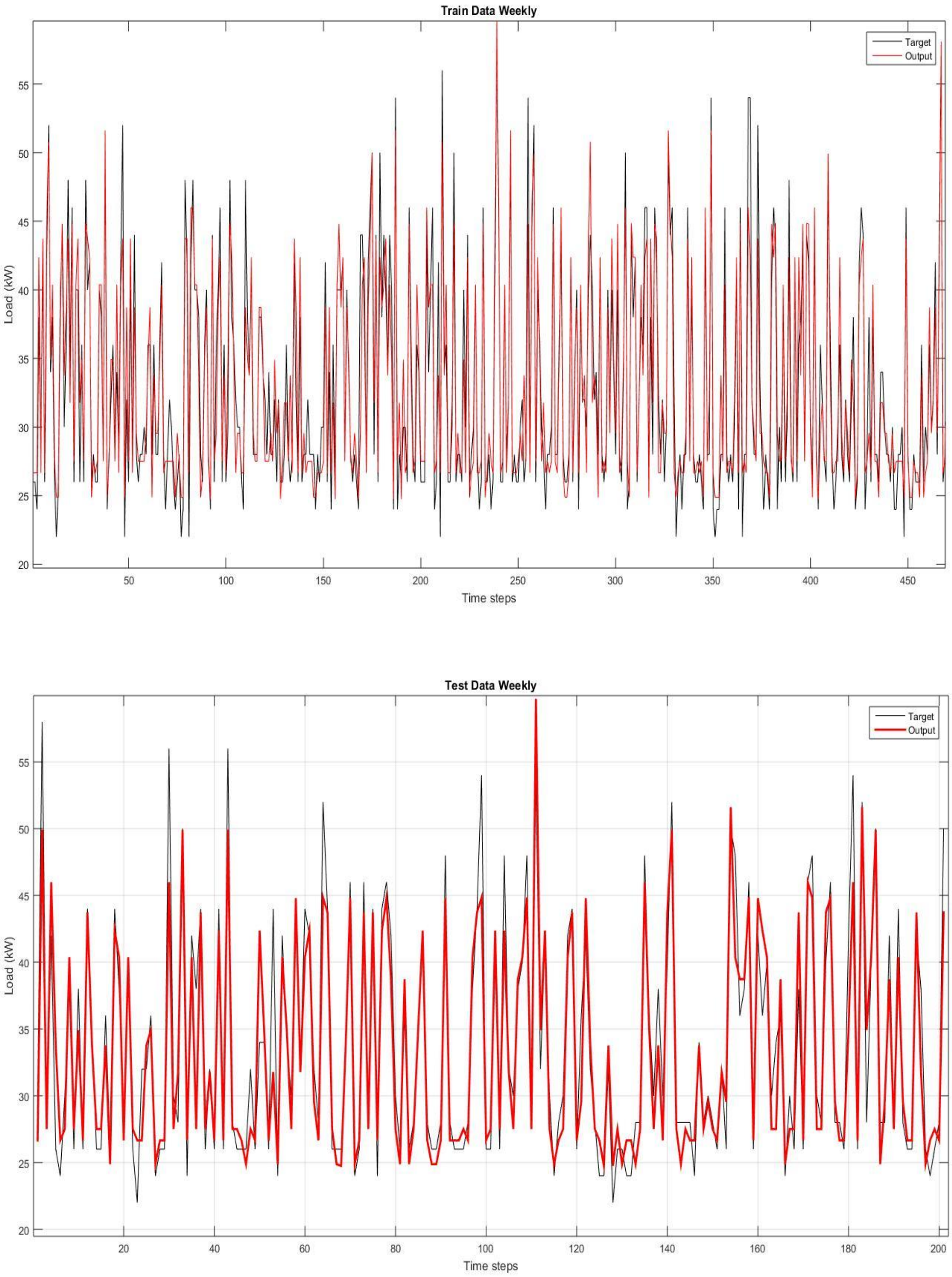
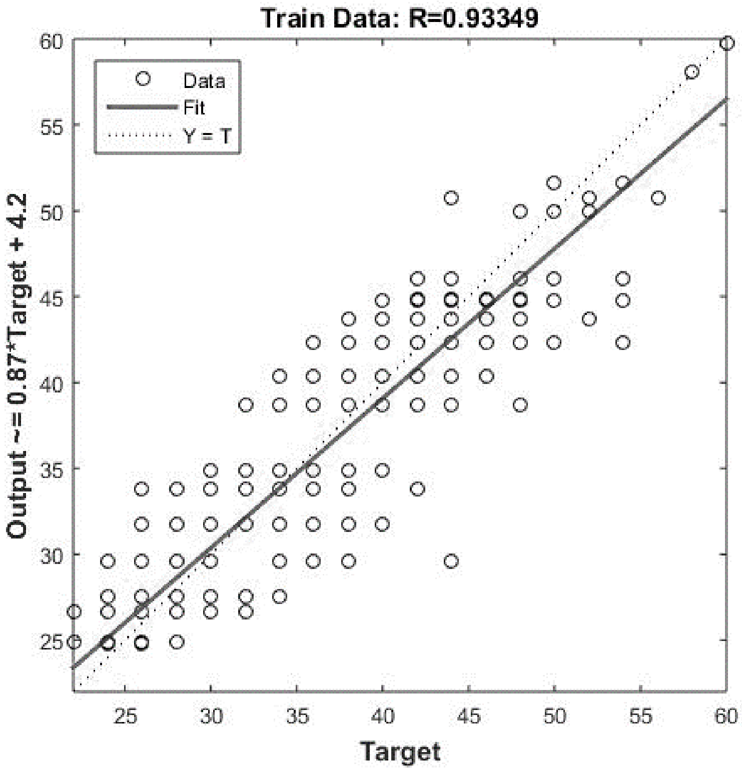


Figure IV-3: Target –Output diagram for train and test data in weekly load prediction



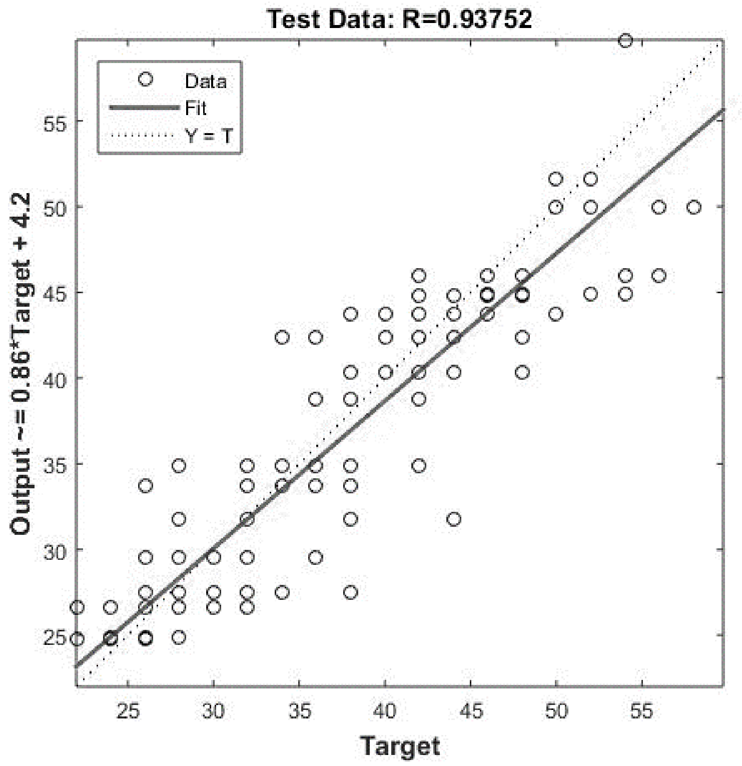


Figure IV-4: Regression diagram for train and test data in weekly load prediction

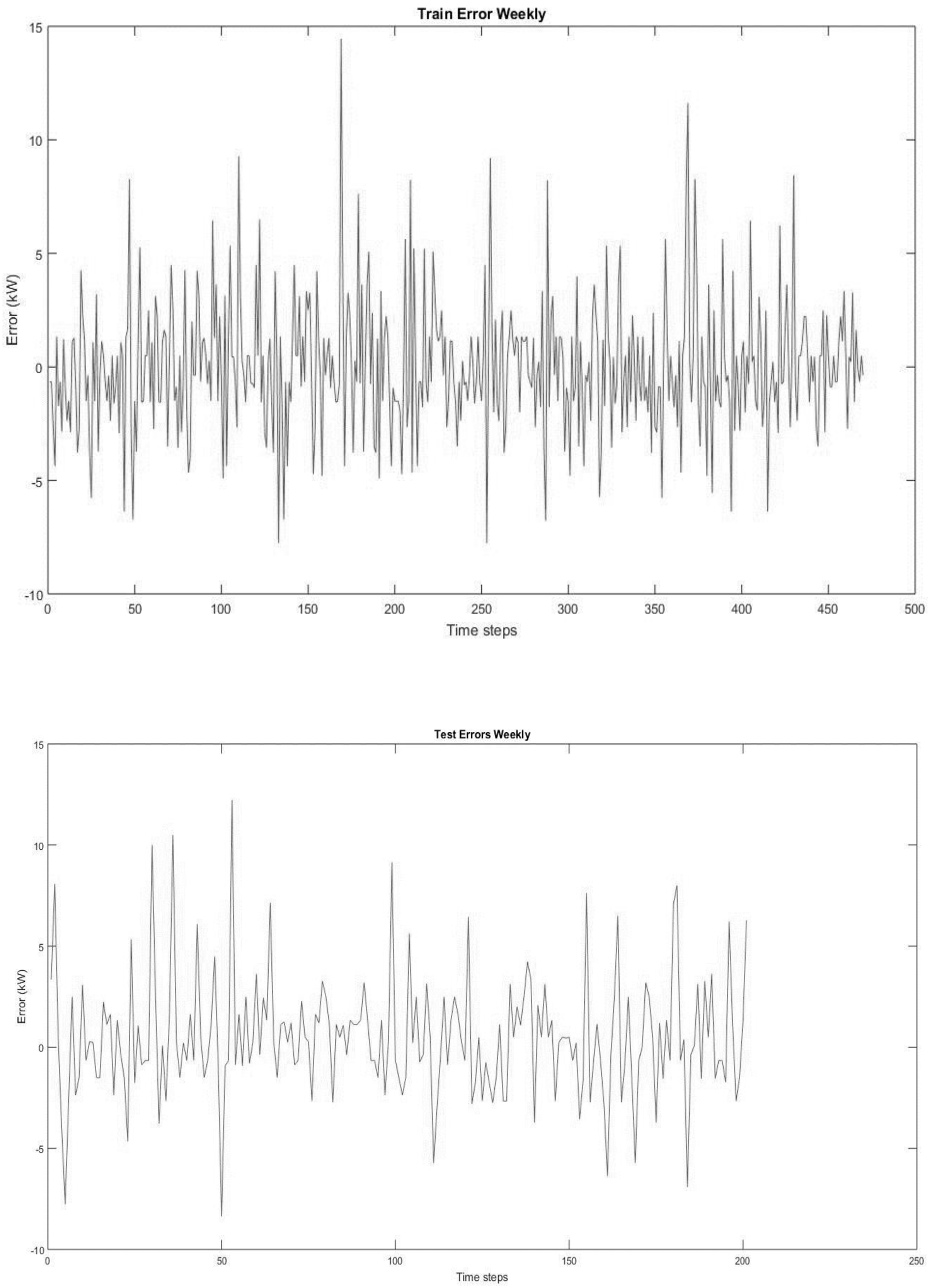


Figure IV-5 Error diagram for train and test data in weekly load prediction

Table IV-1: Weekly prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	3,12e-06	8,2420	2,8709	2,8739	2,1237
Test	0,4616	9,4560	3,0751	3,0478	2,1771

2.2. Monthly Prediction Results:

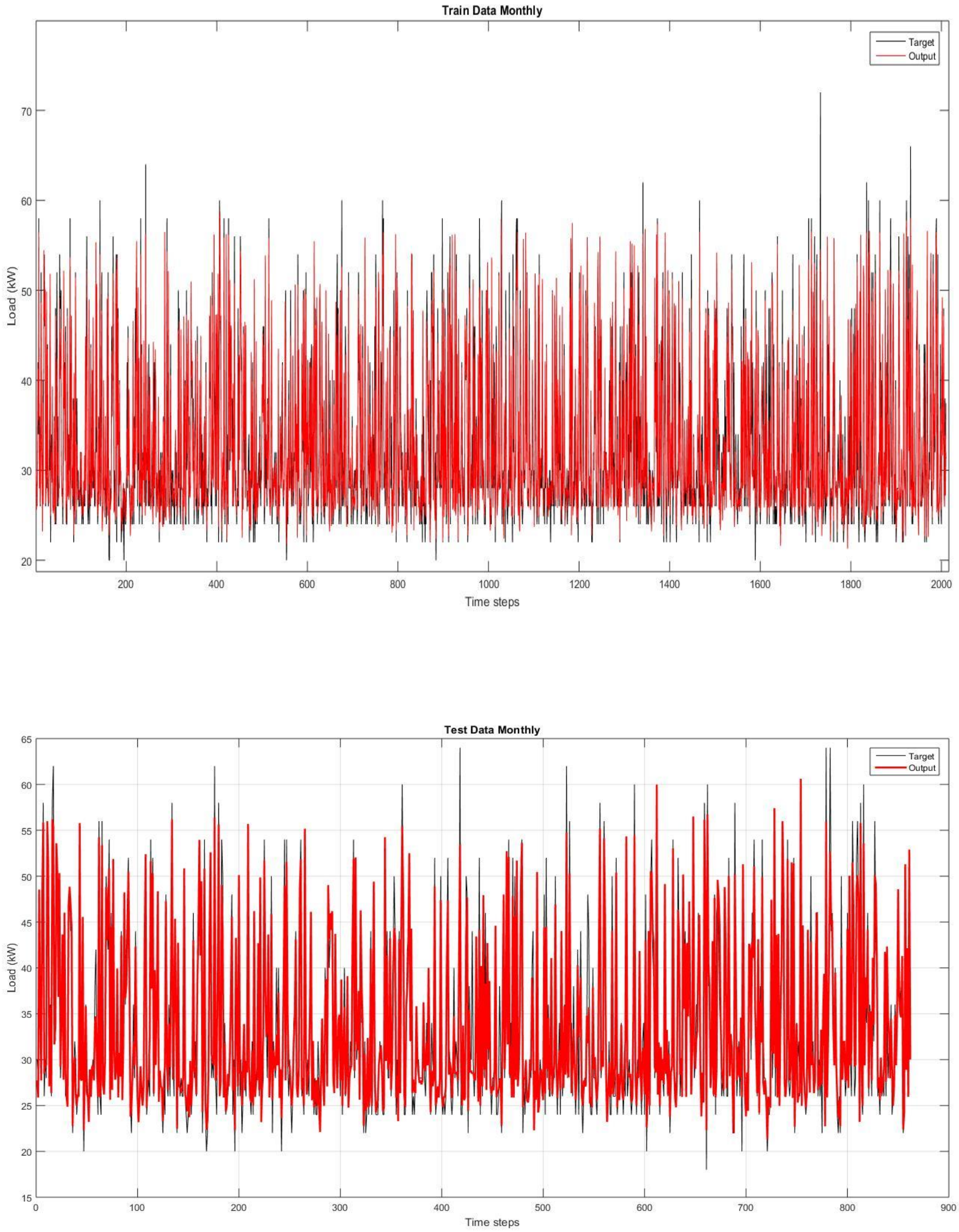


Figure IV-6: Target –Output diagram for train and test data in monthly load prediction

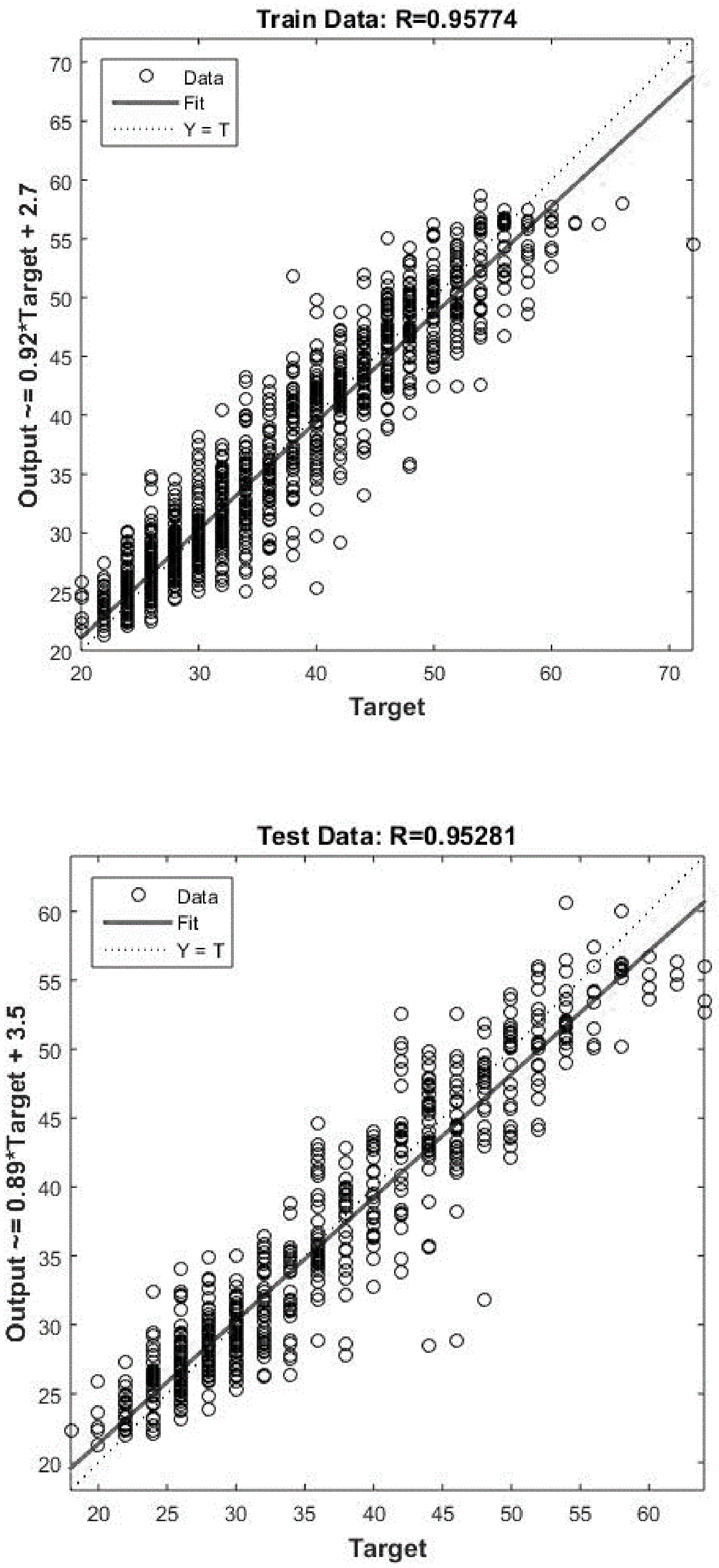
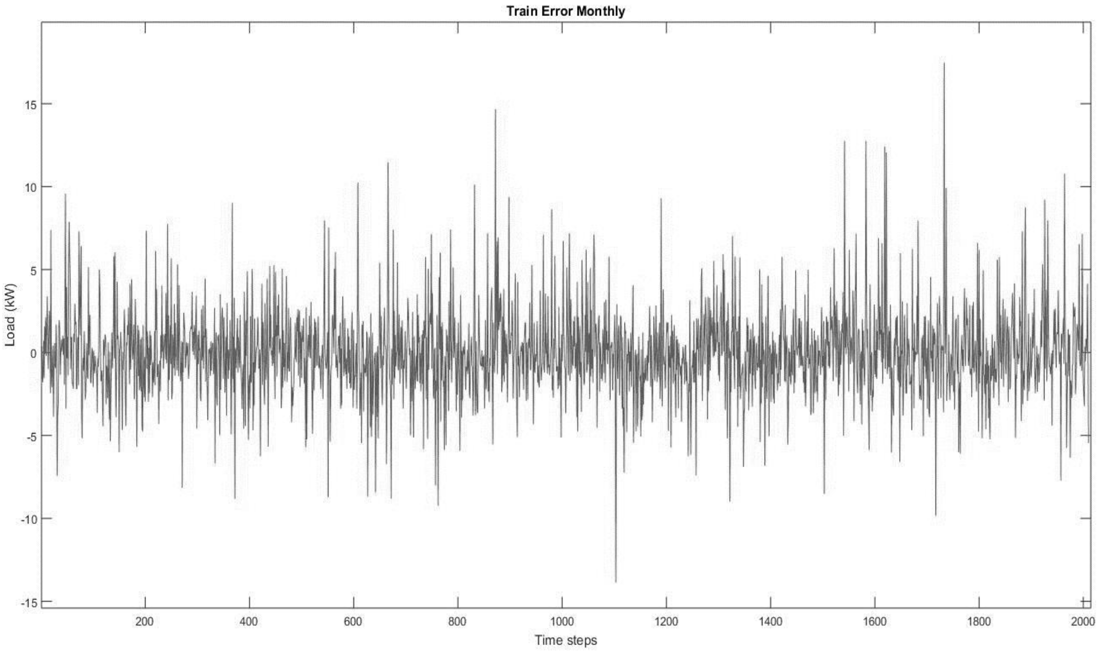


Figure IV-7: Regression diagram for train and test data in monthly load prediction



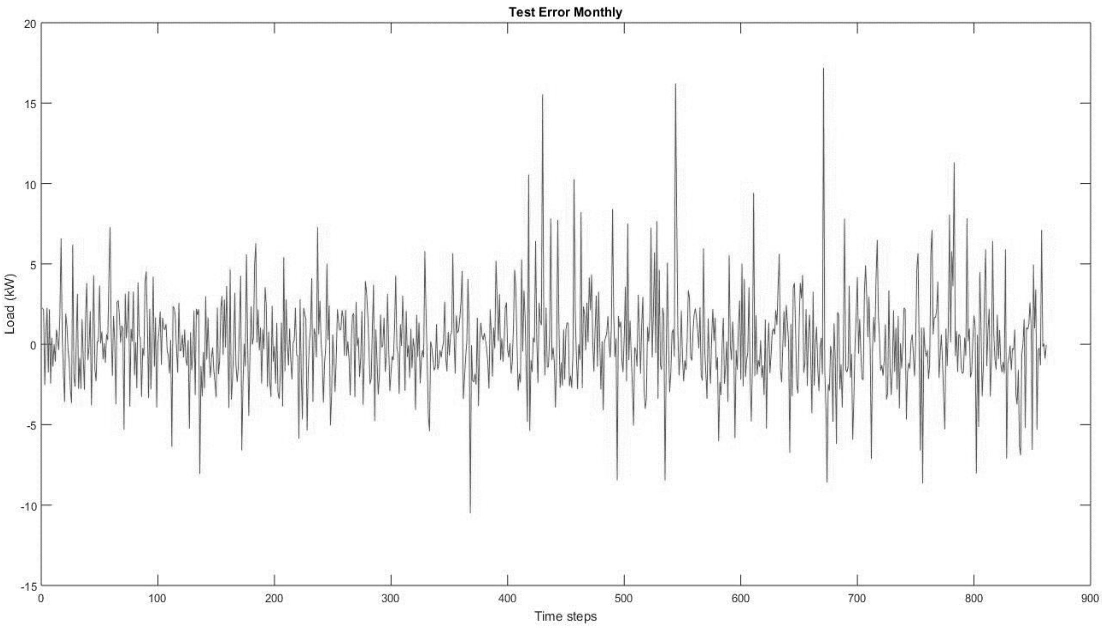


Figure IV-8: Error diagram for train and test data in monthly load prediction

Table IV-2: Monthly prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	2,84e-07	7,2338	2,6896	2,6902	1,9078
Test	0,0874	8,7929	2,9653	2,9657	2,1387

2.3. Seasonal Prediction Results:

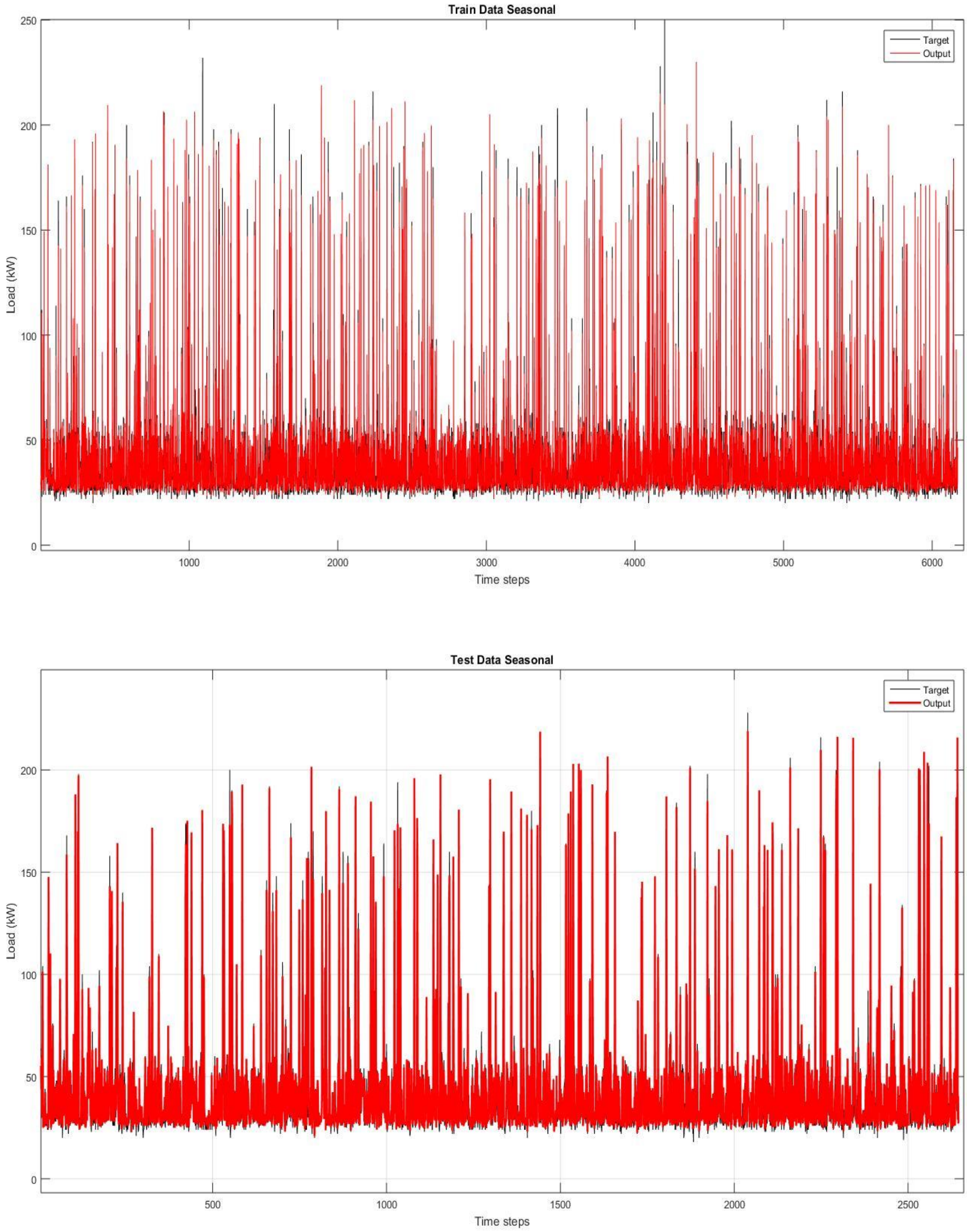


Figure IV-9: Target –Output diagram for train and test data in Seasonal load prediction

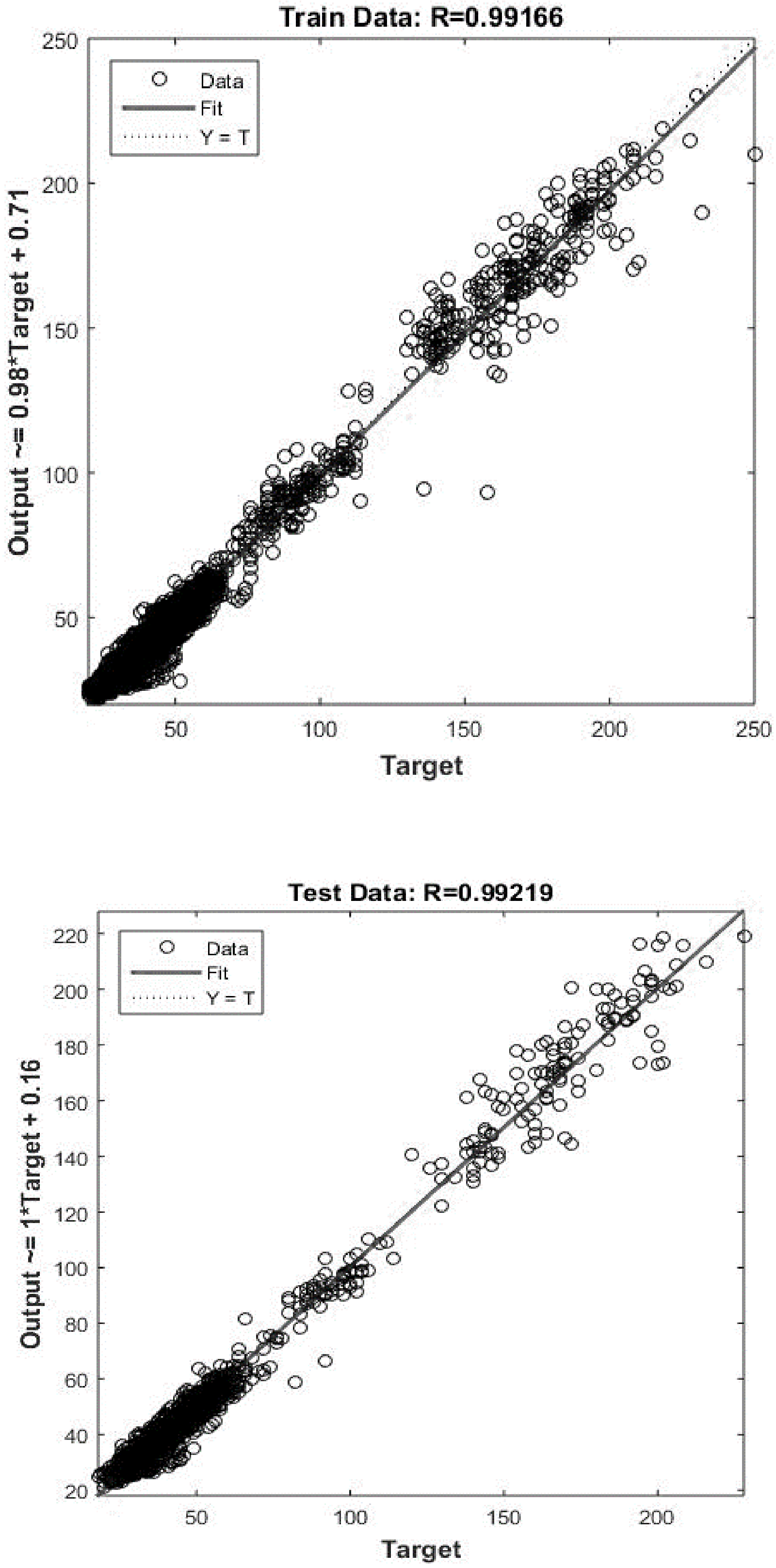


Figure IV-10: Regression diagram for train and test data in seasonal load prediction

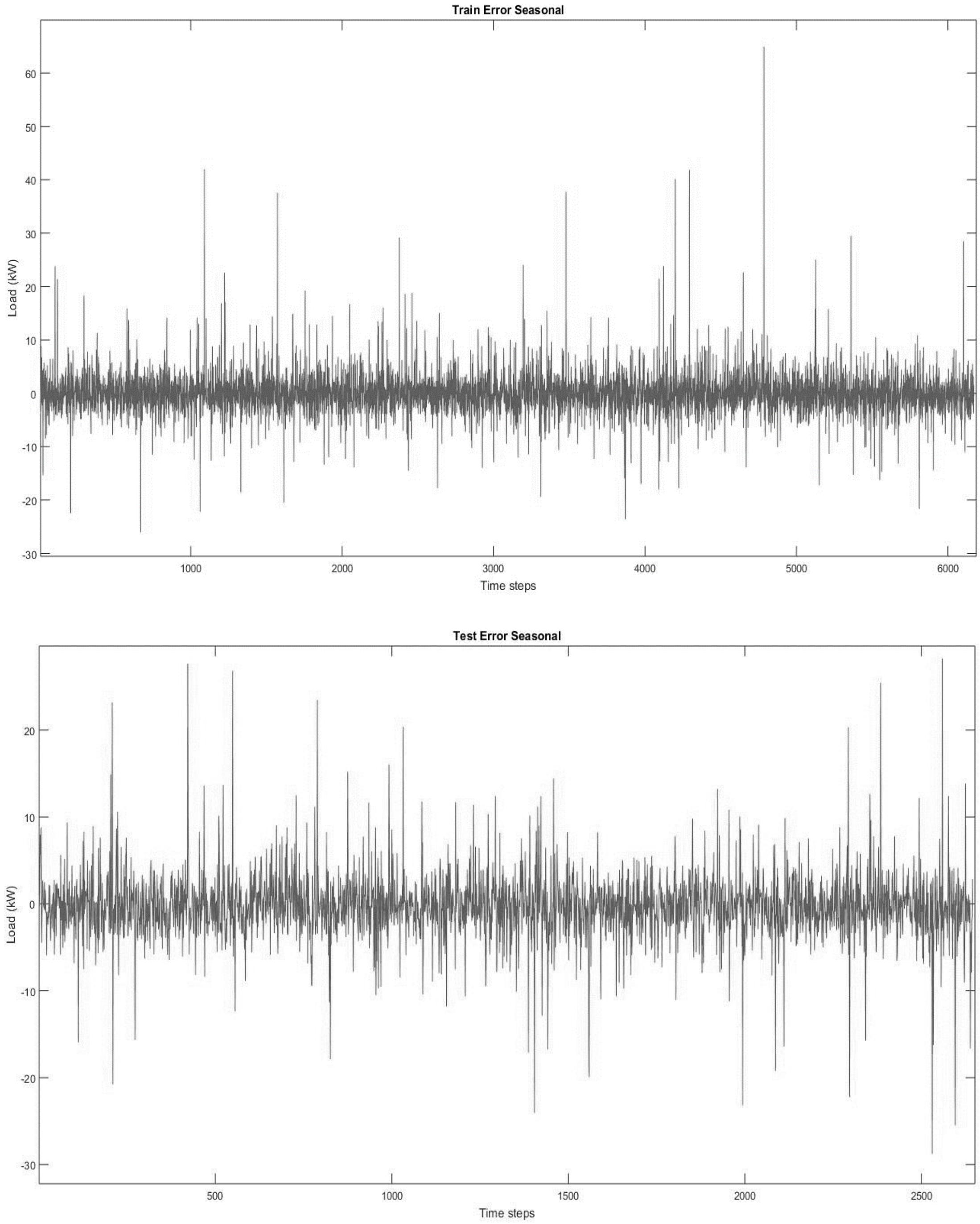


Figure IV-11: Error diagram for train and test data in seasonal load prediction

Table IV-3: Seasonal prediction train and test evaluation

	ME	MSE	RMSE	STD ERROR	MAPE
Train	-2,52e-06	15,1598	3,8936	3,8939	2,5190
Test	-0,2360	15,2277	3,9023	3,8959	2,6051

The output of the proposed model for weekly, monthly and seasonal load forecasting based on ANFIS show very accurate and efficient results in comparison to the common methods in this field. It is also worth to note, that as expected the results suggest that this model presents better performance for a short-term prediction which is in accordance to the other studies.

3.Comparison Between ANN and ANFIS

Comparing the prediction results based on the MLP network presented in chapter III and the ANFIS model discussed in this chapter implies that the both model show excellent performance in short-term applications with less than 2.5% MAPE (mean of the absolute error rate) in weekly prediction which is also reported in many studies as the strong advantage of ANN-based methods. However when the horizon of forecasting is wider, the outcome of the both model tends to be less accurate. Taking only the MAPE into account, there is not a very big difference between the MLP network and ANFIS in monthly load prediction. However, the MAPE alone cannot be a very reliable criterion for assessment. Looking at the test regression plots in both method, it will be clear that the ANFIS has done a significantly better performance in predicting the future load. (With 95% regression rate against 75% in MLP network).

The difference between these two approaches become even clearer when the results of a longer term load prediction is taken into consideration. Here, the regression rate of the ANFIS model is 97% while the regression rate in MLP network is 56%. Also the MAPE in the ANFIS-based method is roughly 2,6 % which is the lowest reported rate in the similar studies with almost the same datasets and prediction horizon. It is also worth to mention that in both cases the input data has been fed to the model systematically based on the chaos and concept drift existence in time series, which itself improves the accuracy of outcome. The table IV-4 compares the results based on this method with some other recent studies in the field of load forecasting:

Table IV-4: Comparison between the result of selected recent studies

Studies	Year	Method	Type	MAPE
Topalli and Erkmen (Topalli AK, 2003)	2001	MLP (online learning)	Short-term	3,8936
Filik et al. (Filik UB, 2011)	2005	ANN	Long-term	5,38
	2005	Mathematical Method	Long-term	8,8
Esener et al. (Esener I, 2013)	2010	EMD and RBF NN	Short-term	2,64
Seema Pal1, A. K. Sharma (Seema Pal A. K., 2015)	2015	ANFIS	24-ahead	5,7
Hasan Huseyin¸ Mehmet Çunkaş (Çunkaş, 2015)	2015	ANFIS	Short-term	1,8 ~ 2,6
Bayram Akdemir, Nurettin Çetinkaya (Bayram Akdemir, 2011)	2012	ANFIS	Mid/Long-term	2,4 ~ 3,7
This Study (based on Chaos, Concept Drift and Systematic feeding of Input)	2018	ANN-Chaos	Short-term	2,3
		ANFIS-Chaos		2,1
	2018	ANN – Chaos	Mid-term	3,7
		ANFIS-Chaos		2,1
	2018	ANN – Chaos	Long-term	21,6
		ANFIS – Chaos		2,6

Since only the electrical load data of one year (2014) was taken into account for the training process, the proposed model was not tested for a yearly load prediction. Nevertheless, the method and the implemented model is flexible and applicable to any time horizon and the author believes, that a longer time series (for example over two or more years) may result in even better and more accurate outcomes.

V. Chapter 5

1. The Aim of Energy System Modelling

The concerns about the increased misuse of natural resources and shortage of them in near future and also the destruction and pollution of environment due to the expansion, construction and operation of energy systems have generated a growing interest in the improvement of energy efficiency. In other words, the new energy systems should be designed in a way by which, the consumers' needs are fully satisfied while natural resources exploitation and environmental pollutions are minimized through efficient implementation of renewable energy systems.

On the other hand, the energy production from renewable energy sources and also the distributed energy generation systems based on them are gaining more importance every day. The increasing rate of renewable energy sources is reported about 2.5 percent per year in the world. (Administration, 2014) According to the German Association of Energy and Water Industries (BDEW), the electricity produced from renewable energy sources has grown in 2016 again and was about 31.7 percent (381 Terawatt hours) of gross electricity generation in Germany. Due to this rapid development of energy systems based on renewable resources and their facilities and equipment, such as photovoltaics and wind turbines, new challenges have arisen such as the fluctuating electricity production, production dependency on seasonal and local situations or the decrease in electricity consumption at the times with the maximum electricity generation capacity. There have been many studies and projects in the last decades to find a solution for such problems. Introduction of new methods for energy storage including new ideas for heat storage or power to X projects were some of the most important developments in this field. Therefore, different optimization methods and procedures have to be tested and applied on the systems to achieve the best combination in modern cross-sectoral energy systems. To perform such tests, a profound understanding of an energy system mechanism and its modelling procedure is indeed very important and necessary. The simulation runs of the model provides us with valuable information such as system responses to different changes and the appraisal of optimization methods, prior to real-life application of such methods to real systems.

Energy system modeling has a very long history and has always been an interesting subject for international energy agencies and administrations all over the world. The variety of these models change from large-scale mathematical models for long-term energy projections to simpler and smaller models for specific applications.

There are five main areas in energy system modeling, namely:

- Energy production
- Energy transmission
- Energy consumption
- Energy economics
- Environmental impacts of energy production or consumption

Based on the application domain, one or more of the above points play the key role in an energy system model. For instance, where the financial detail of energy production is important, an economic-based model can be very helpful, while an environmental-based model can be of a higher priority, where the pollution is the primary concern. (Bri-Mathias S. Hodge, 2011)

Another classification of energy system simulations can be done based on their approach toward modeling. Generally the energy system models can categorized into top-down, bottom-up or Hybrid models. The top-down models focus on aggregated quantities and include a more general overview. On the other side, the bottom-up models represent more detailed illustration of a system along with its detailed components. The Combination of these two approaches include the advantages of both methods. (Wei, 2006)

For Modelling the Energy Park in CUTEC institute as the case study of this dissertation the bottom-up approach was chosen, as the simulation of each component in Energy Park was important. In this way the model is very flexible for defining different specifications in each unit and also it is capable of further development for future studies or in case of adding new components to the system. Furthermore, the effects of each unit in the generation process in Energy Park is easy to follow in detail with this approach.

2. Simulation Model of a Decentralized Energy System

As the name suggests, a decentralized energy system is a distributed energy production field close to the consumption area, instead of a large power plant which transmits the electricity or heat through the national grid.

Aside from addressing the needs of almost 1.5 billion people lacking electricity around the world, the local generation of energy can reduce the transmission losses, while simultaneously facilitating the transitioning toward a decarbonized energy system. The statistics show that the global energy system in the world is facing a unique era where the decentralized energy networks are rapidly spreading, based on super-efficient end-use appliances and low-cost facilities, which offer high security of supply, where the consumers of a specific area do not have to share a supply or rely on relatively few, large and remote power stations. In addition to that, distributed energy systems provide economic benefits too. In a long-term period, a decentralized energy system can offer more competitive prices than the traditional energy generation units. Although the initial costs may be higher, a special decentralized energy tariff creates more stable pricing. For governments and energy organizations, a decentralized energy system is a cost-effective route to achieve carbon targets and establish a reliable and flexible energy generation with wide area of applications through locally provided, sustainable, competitive and smarter energy choice.

An effective method to analyze the distributed energy system is modeling and simulation by taking different constraints into account. This simulation model should contain the main energy generation units and their features, energy storage systems and also the consumption loads. By modeling such an energy system and applying a precise electrical load forecast (for example based on the method introduced in the previous chapters) as input, the best combination of the generating units based on their priority can be determined and the operation schedule of each unit can be defined for future planning.

Figure V-1 shows the concept of a decentralized energy system simulation model:

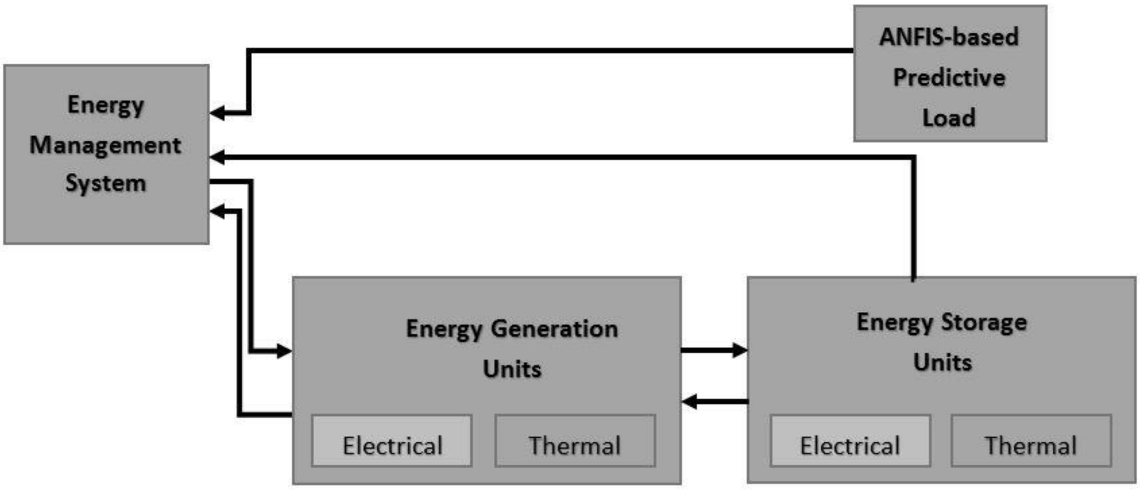


Figure V-1: Main parts and interactions of a distributed energy system

As shown in the above figure, the electrical (or heat) load will be predicted based on the ANFIS model and it will be fed into the Energy Management System (EMS), which can vary from a simple gas regulator to a very complex smart management system. The generation information of the renewable units such as photovoltaics or solar collectors will also be delivered to the EMS and based on the residual load, the EMS decides which units to start up. Furthermore, it defines the level of their power output to supply electricity and heat. The generation surplus if any, will be fed to the energy storage units, which can also supply a part of the load in the peak hours.

The next part introduces the energy system model developed for the case study of CUTEC Energy Park with all its components. It also provides some sample results for a typical hot and cold week of the year from the simulation of the year 2014.

3.Case Study of Energy Park in CUTEC

3.1. Introduction of Energy Park CUTEC

The Energy Park in CUTEC institute is a model system in form of a laboratory and experimental/pilot plant which provides the possibility of a gradual construction for new units and developments for future projects. Since there are several other industrial pilot units in CUTEC, It is possible to experiment different scenarios based on the electricity and heat consumption. For a comprehensive observation and modelling of a future decentralized energy system, a cross-sector combination of electricity, heat and certain substances production were considered in designing the energy park. Electrical and heat storage systems were also included in energy park to serve as a backup for the system in times of high demand or to store the energy surplus in times of lower demands.

The energy park is connected to the grid and based on the contract with the municipal utility company in Clausthal-Zellerfeld, the price of each kWh is constant for all the hours of a day and the excess electricity from CUTEC (if any) , which may be fed back to the network, will not be refunded. This makes the energy park model to be designed based on the priority in usage of different units and not economic optimization.

The list of all the units, which were considered for modeling in the energy park is shown in the table V-1:

Table V-1: Units in CUTEC Energy Park Model

Unit	Electrical Power (kW)	Thermal Power (kW)	Capacity	Remarks
Boiler (Buderus)	-	650	-	-
CHP[5] (Buderus)	70	109	-	capable of modulating
CHP (Powertherm)	20	43	-	capable of modulating
CHP (Senertec Dachs)	5,5	12,5	-	-
Solar Collectors	-	-	-	59 m^2 total area
Photovoltaics	-	-	-	70,98 m^2 total area
Battery Storage	-	-	120 kWh	Lead battery
Latent Heat storage	-	-	2,3 MWh	-
Cooling system	-	-40	-	10 kW cold water power

[5] Combined Heat and Power production unit

These units were modeled and implemented in MATLAB and MATLAB Simulink. The next part will provide a short introduction to MATLAB software and Simulink environment. After that a brief explanation of each unit as well as the other important parameters in Energy Park model will be given.

3.2. MATLAB and MATLAB-Simulink

MATLAB / Simulink: MATLAB (MATrix Labratory) from The MathWorks Inc. is a program for the numerical calculation and it used for the analysis and development of control systems.

Many functionalities in MATLAB can be accessed by so called toolboxes. MATLAB also provides user interfaces and interfaces for programming in other languages, such as C, C ++, Java and FORTRAN. Most of the functions and conventions are logical and easy to use for numerical calculation. In comparison to other computer algebra programs, MATLAB requires basically just one data structure and that is the matrix.

MATLAB consists of five main components:

1. ***Application environment***:
   The application environment allows the creation of the program in order to establish effective communication with the program. For this purpose powerful graphical interfaces are used, such as command window, command history window (Command History) etc...

2. ***Library of mathematical functions:***
   These libraries include a variety of computational algorithms of basic functions (such as sine and cosine) and complex mathematics to advanced features (such as inverse matrix, Bessel functions, and fast Fourier transform).

3. ***Programming language:***
   The software has a higher programming language and array / matrix control commands, so that it allows the adoption of different data structures and getting input / output functions and it makes the object-oriented programming possible. The programming in

this environment is possible for large programs (Large Scale) or even small programs (Small Scale).

4. ***Graphical properties:***

   The software provides the ability to view matrices and vectors legibly. Besides, the two and three dimensional graphs, animations and representations of images in the preferred format can be shown. Of course there is also the ability to develop graphical interfaces by yourself.

5. ***Interfaces with the external environment:***

   MATLAB has bi-directional communication with FORTRAN and C. The functions can be accessed in three ways:

   - Calling MATLAB programs in C software in the form of .dll.
   - Use of MATLAB as computing machine.
   - Writing and reading of M-files (MATLAB files) (Tranquillo, 2011)

Figure V-2 shows an example of a MATLAB code:

```
function [m, b]= linreg( x, y )
   medx = mean(x);                                          % arithmetisches Mittel von x
   medy = mean(y);                         % arithmetisches Mittel von y
   N = size(x,2);
   num = 0;
   den = 0;
   for i = 1:N                             % Summe von i bis N
      num = num + (x(i)-medx)*(y(i)-medy);  % Zähler
      den = den + (x(i)-medx)^2;            % Nenner
   end
   m = num/den;
   b = medy-m*medx;
end
```

Figure V-2: A Sample Code in MATLAB

Simulink is a MATLAB tool for modeling, analysis and synthesis of linear and nonlinear dynamic systems. In Simulink environment, the system can be modeled by means of blocks and then executed and the results will be shown. The parameters can be changed easily and directly in the simulation model. Simulink is capable of communicating with most programming environments. (Florian Grupp, 2007)A Simulink model normally consists of three areas:

- Inputs
- Blocks for system modeling
- Outputs

Figure 7.2 shows a Simulink model example of the elementary transfer elements (blocks) P, I, PT1 and PT2. As the input a sinusoidal signal with constant amplitude and frequency has been taken. The results of the simulation is shown in the figure 7.3. (Hashemifarzad, 2014)

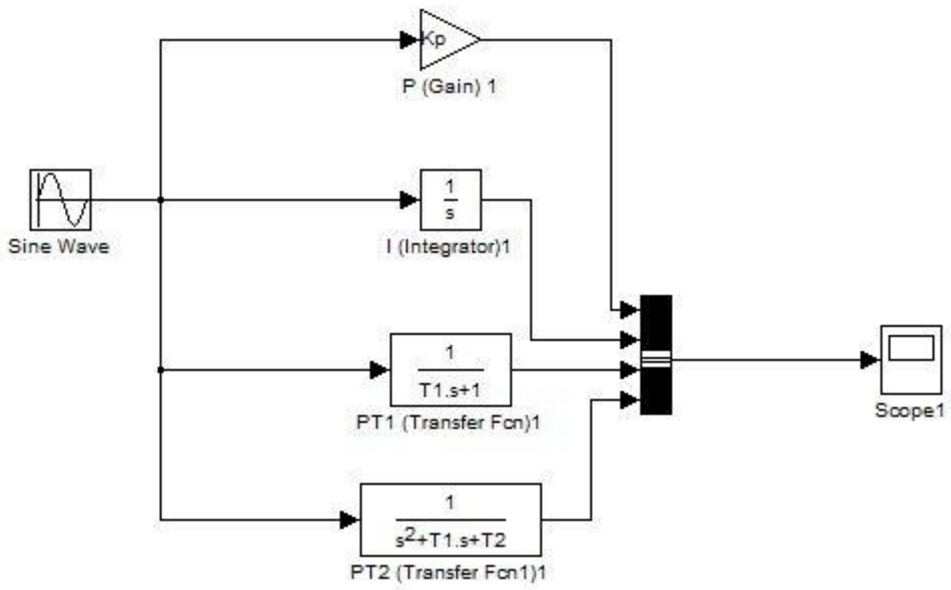


Figure V-3: A Simulink model with different transferring blocks

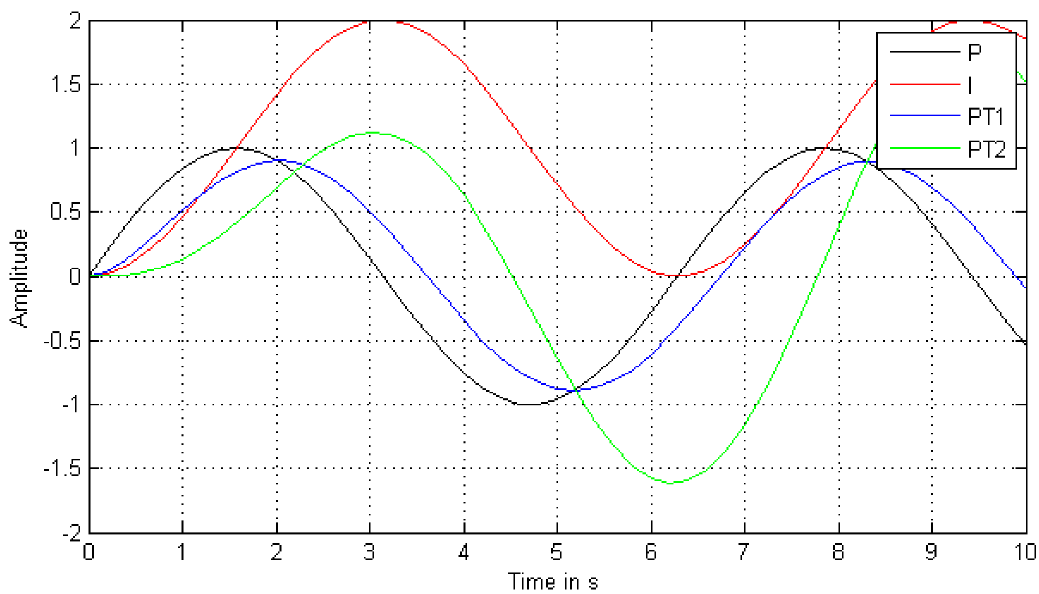


Figure V-4: Sinusoidal responses of different transmission elements

3.3. Model

The Energy Park model was implemented in MATLAB. The Model was designed in such a flexible way that can be applied for any micro gird with similar units and the parameters of each unit can be defined by the user separately. It is also possible to feed the electrical and thermal load and also the solar irradiation in form of time series. The user can also add or remove some of the units based on his interest to test different combinations of generation units. As mentioned in the part 3.1. the electricity supply is connected to the grid and based on the contract with the municipal utility company in Clausthal-Zellerfeld, the price of electricity usage is constant and the fed-back electricity to the network will not be refunded. As a result an economic optimization was not the intention of this model and the look up tables for the refunding were not needed in the model. The strategy for energy supply (electrical / thermal) of CUTEC was thought to be done by the solar collectors and photovoltaics and also storages in the first place, as far as available. The residual last would then have to be covered by the other units, for which the user can define the priority.

This general model of a micro grid contains different parts. In the following, the main parts in the CUTEC Energy Park will be described as the case study:

Gas Regulator

The gas regulator is the main controller in the model. It regulates the gas consumption for the boiler and all the CHP units based on the electrical and thermal load. The system is electric driven, meaning that covering the electrical load of the system has priority over thermal load. The gas regulator module in the model needs the following input values to control the system:

- The current generation level of different units
- Electrical residual load
- Thermal residual load
- Storage filling level

All these values are transferred in form of per unit. The output from the gas regulator will be sent to the first CHP in the order of priority and the level of the generation of this unit will be determined, if the first CHP is not able to cover all the residual last, a signal will be sent to the next CHP in the order of priority and this goes on to the last CHP available. This signal also defines the level of generation in CHP units with the capability of modulation. In case of electricity surplus, the batteries will be charged, otherwise the rest of lacking electricity in the system will be provided by the network. If there is still thermal energy needed in the system a signal will be sent to the boiler to generate the needed extra heat.

Boiler

The Buderus boiler in Energy Park has a nominal thermal power of 640 kW with 50/30 °C input/output flow and 585.4 kW with 80/60 °C input/output flow. Based on the designed strategy the boiler will cover the residual heat last and also when the heat storage level is less than 10%. The boiler receives a signal from the gas regulator when one of the above situations happen. All the signals in the model transfer in “per unit” mode. This per unit signal will then be multiplied by the nominal power of the boiler and specifies the output of boiler.

CHP Unit- Buderus

The combined heat and power (CHP) generation unit from Buderus Company is the newest element installed in the energy park. The installation was completed in 2016. The nominal electrical power of this unit is 70 kW with the possibility of modulating with the minimum generation power of 40 kW. The nominal thermal power is 109 kW. Based on the technical data sheet, the electrical efficiency of this unit is 32 percent, where the thermal efficiency is around 51 percent, which makes an overall efficiency of almost 83% at the best conditions. This module receives a controlling signal from the gas regulator in per unit form when more electricity generation is needed. In the Simulink model, this block has two outputs which are the electrical and thermal generations. The other CHP modules in the model implementation also have the same types of inputs and outputs. (Figure V-5)

CHP Unit- Powertherm

The second CHP unit in the energy park is the Powertherm with a nominal electrical power of 20 kW and thermal power of 43 kW. This unit is also capable of modulating between 14 kW to

20 kW with the steps of 2 kW. The electrical efficiency of this unit is around 23 percent and the thermal efficiency is almost 52 percent, which reaches an overall efficiency of 77 percent.

CHP Unit- Senertec Dachs

The smallest and oldest CHP in the energy park is one from Senertec Dachs Company with a nominal electrical power of 5.5 kW and thermal power of 12.5 kW. This model of CHP is normally installed for households in Germany. The electrical efficiency of this small CHP is around 28 percent and the thermal efficiency is 52 percent.

Photovoltaics (PV)

A PV system is also used for the power supply of the energy park. The PV panels are installed on and around the main entrance of the CUTEC building with a total area of 70.98 m^2. There are two main similar blocks with surface azimuth angle of 6.1° SW and tilt angle of 60°. Since the PV panels are relatively old the efficiency of them is about 12% and there will be a shading loss of almost 16%. All of these factors were taken into consideration in the implementation of the PV unit in the energy park model. The user can add the solar irradiation of the region in forms of times series and based on that the block will calculate the electrical production of this unit. For the case study of CUTEC, the solar irradiation data from a near measurement station in Osterode city was chosen. The irradiation data were measured and recorded hourly.

Solar Collectors

The Solar collectors are installed on the roof of one of the halls in CUTEC in three similar blocks with a total area of 60 m^2. The implemented module in MATLAB calculates the output power of these blocks based on the same solar irradiation for PV panels and the coefficient of absorption correction factor provided by the Ritter Solar company.

Heat Storage

To store the excess heat from the system and save it as a back-up for the overhaul times, a latent heat storage unit with a capacity of 2.3 MWh was integrated in the energy park. This storage is filled with Sodium Acetate Tri-hydrate (CH3COONA) with a molar mass of 136.08 g/mol °K and specific heat capacity of 229 J/mol, which has a high heat of fusion and is considered as a very good **phase change material** (PCM). Such substances can melt and solidify at a certain temperature, and so they are capable of storing and releasing large amounts of energy. Heat is absorbed or released when the material changes from solid to liquid and vice versa. In case of Sodium Acetate Tri-hydrate, the melting and solidifying temperature is around 59° C.

Battery System

There are 184 lead battery blocks in CUTEC from BAE Company. For the last project, these blocks are divided into two main parallel strands. The nominal voltage of each block is 6 V with a nominal capacity of 225 Ah, which provide a total capacity of almost 120 KWh.

Electrical / Thermal Load

The user can add the historical time series of electrical or thermal load from the user interface. In the case study of CUTEC, the electrical and thermal load from the year 2014 were taken into consideration. The time resolution of electrical load data is 15 minute. The thermal load of CUTEC in 2014 was measured hourly.

Utility Price

The user can also add the price of electricity and natural gas in order to get an approximate final system costs in results.

A user interface was designed with MATLAB graphical user interface (GUI), which enables the user to specify the features of different units and select the electrical and thermal load time series and also the irradiation data. Figure V-5 shows the main user interface and figure V-6 shows the interface to enter specific features of each CHP unit.

Figure V-5: Main user-interface

The detailed implementation of each module in MATLAB Simulink and also the related MATLAB codes for this model can be found in the appendix B.

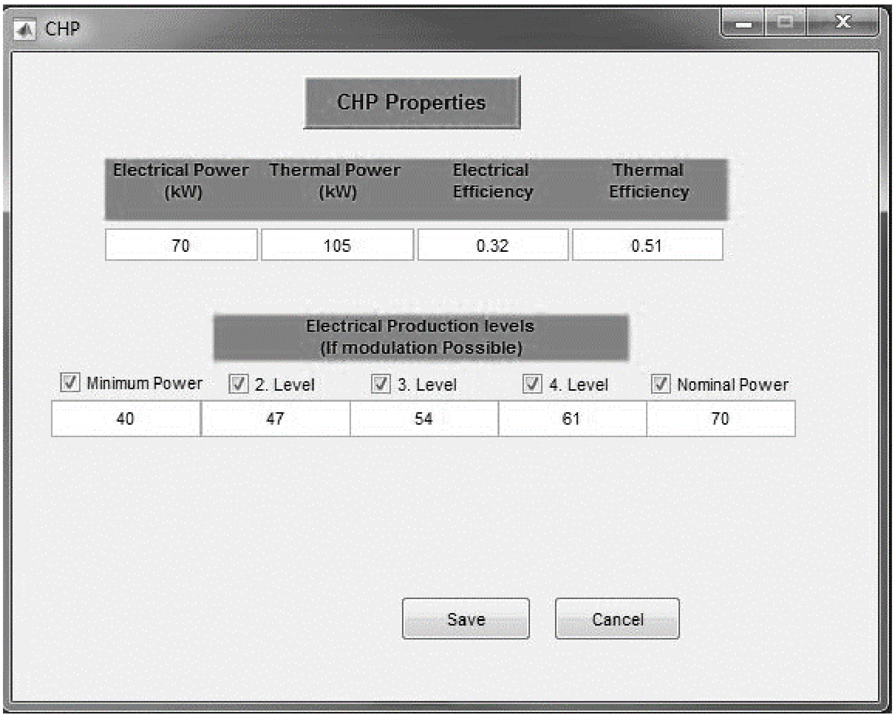


Figure V-6: User interface for CHP units

Outputs:

The implemented model in MATLAB receives all the above mentioned parameters and performs a simulation which provides the following results:

- Cumulative electrical production diagram (With specific color for each unit - 15 min resolution) - Figure V-12
- Cumulative thermal production diagram (With specific color for each unit- 15 min resolution) - Figure V-12
- Status of Battery/ Heat storage (15 min resolution) - Figure V-12
- Total electrical production (together with overall percentage) of each unit - Figure V-9
- Total thermal production (together with overall percentage) of each unit - Figure V-9
- Electrical load vs. total production diagram - Figure V-10
- Thermal load vs. total production diagram - Figure V-11
- Total costs of purchased electricity and natural gas - Figure V-8
- Electrical production plan of each unit in form of Excel data
- Thermal production plan of each unit in form of Excel data

Figure V-7 shows the user interface, when the simulation is completed and the user can choose the type of desired results:

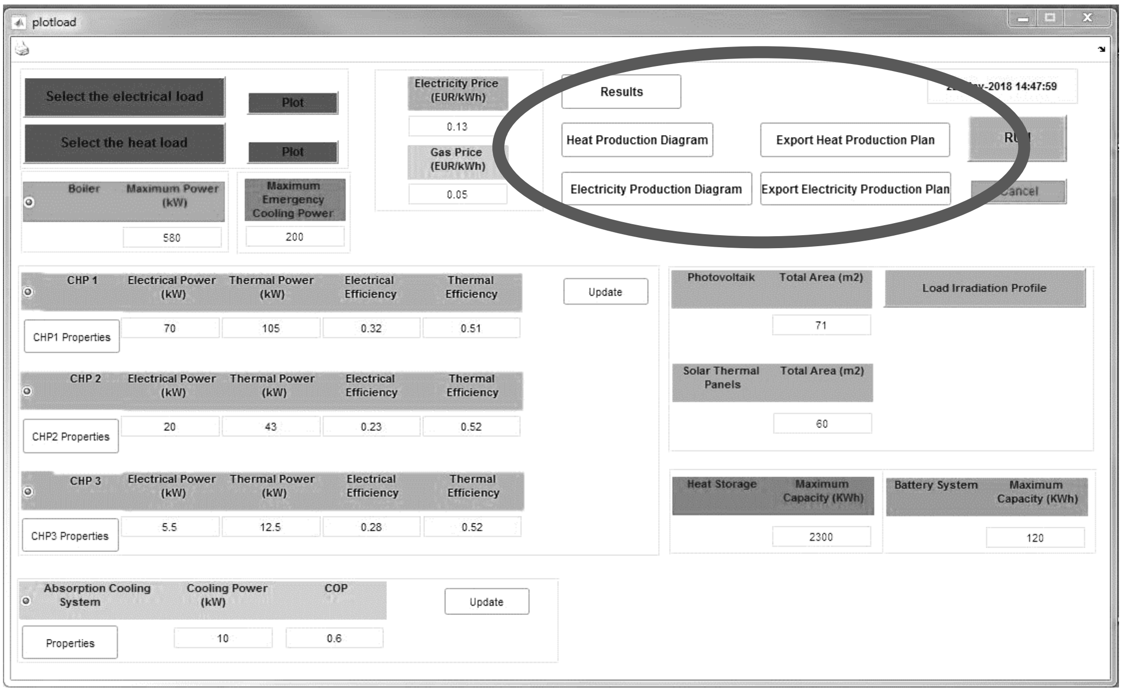


Figure V-7: User interface after completed simulation

The results window provide the detailed production information for each unit as shown in Figure V-8:

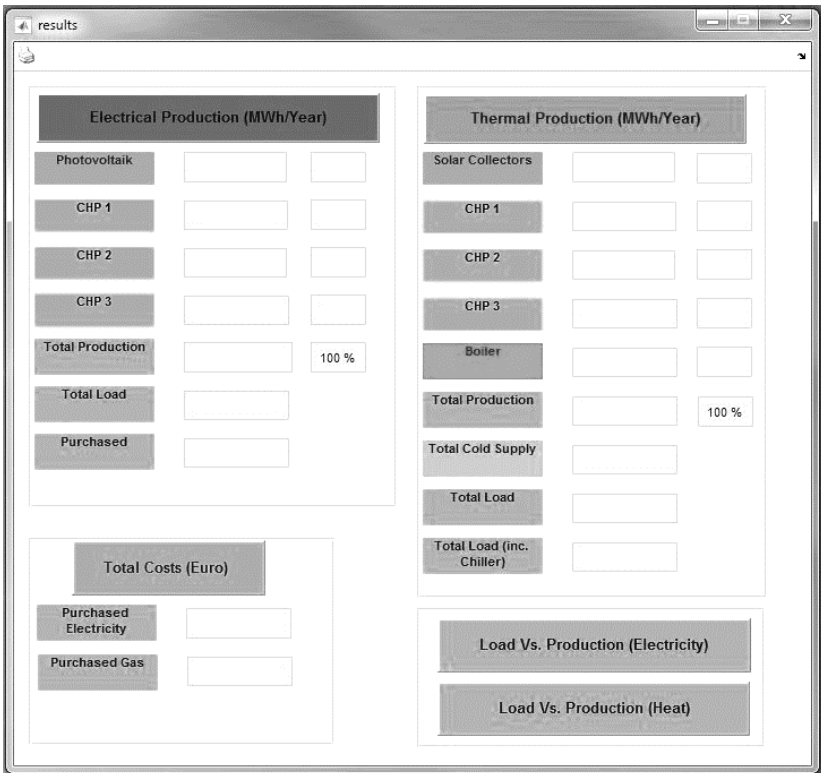


Figure V-8: The results window

3.4. Sample Results

As mentioned in the part 3.3 for the case study of CUTEC Energy Park, the historical time series of electrical and thermal load of CUTEC institute were fed to the model and all the features of the user interface were set to match the units in the energy park. The figures V-9 to V-12 show the main graphical results of this simulation for the sample year 2014 and the detailed generation of each unit as well as the detailed production diagram can be found in the appendix C.

3.4.1. Overall Results – Year 2014

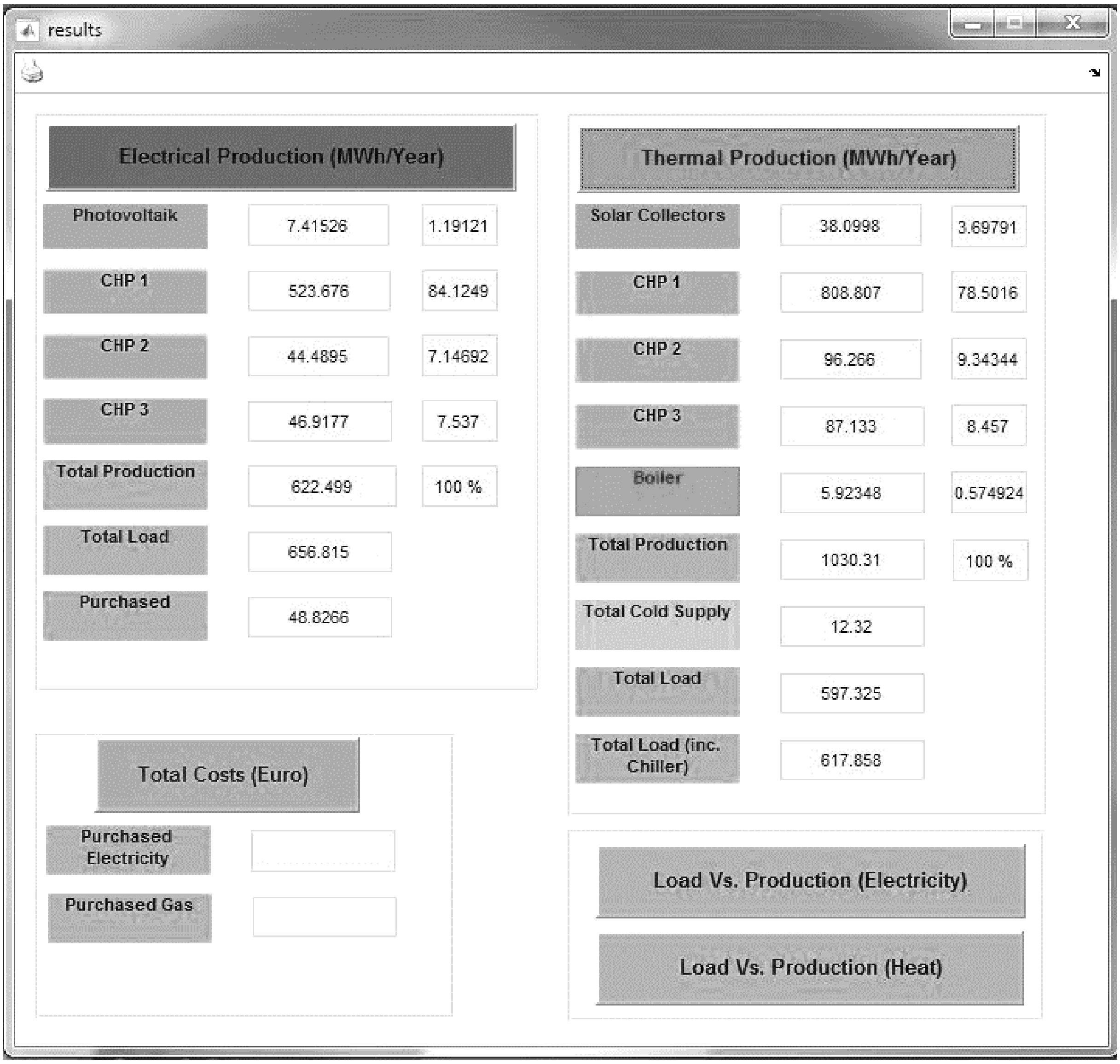


Figure V-9: Overall results

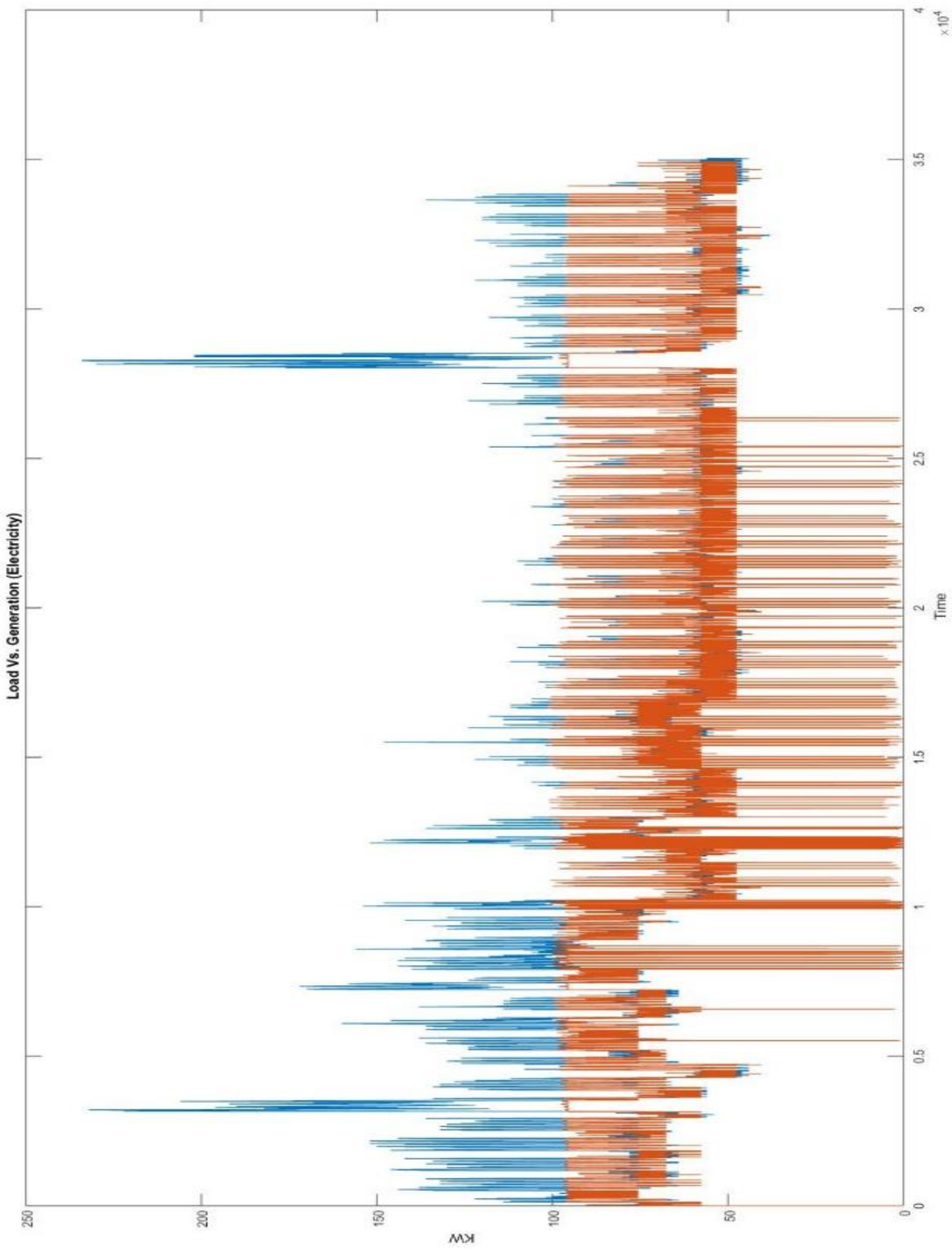


Figure V-10: Electrical Load vs. Generation - 2014

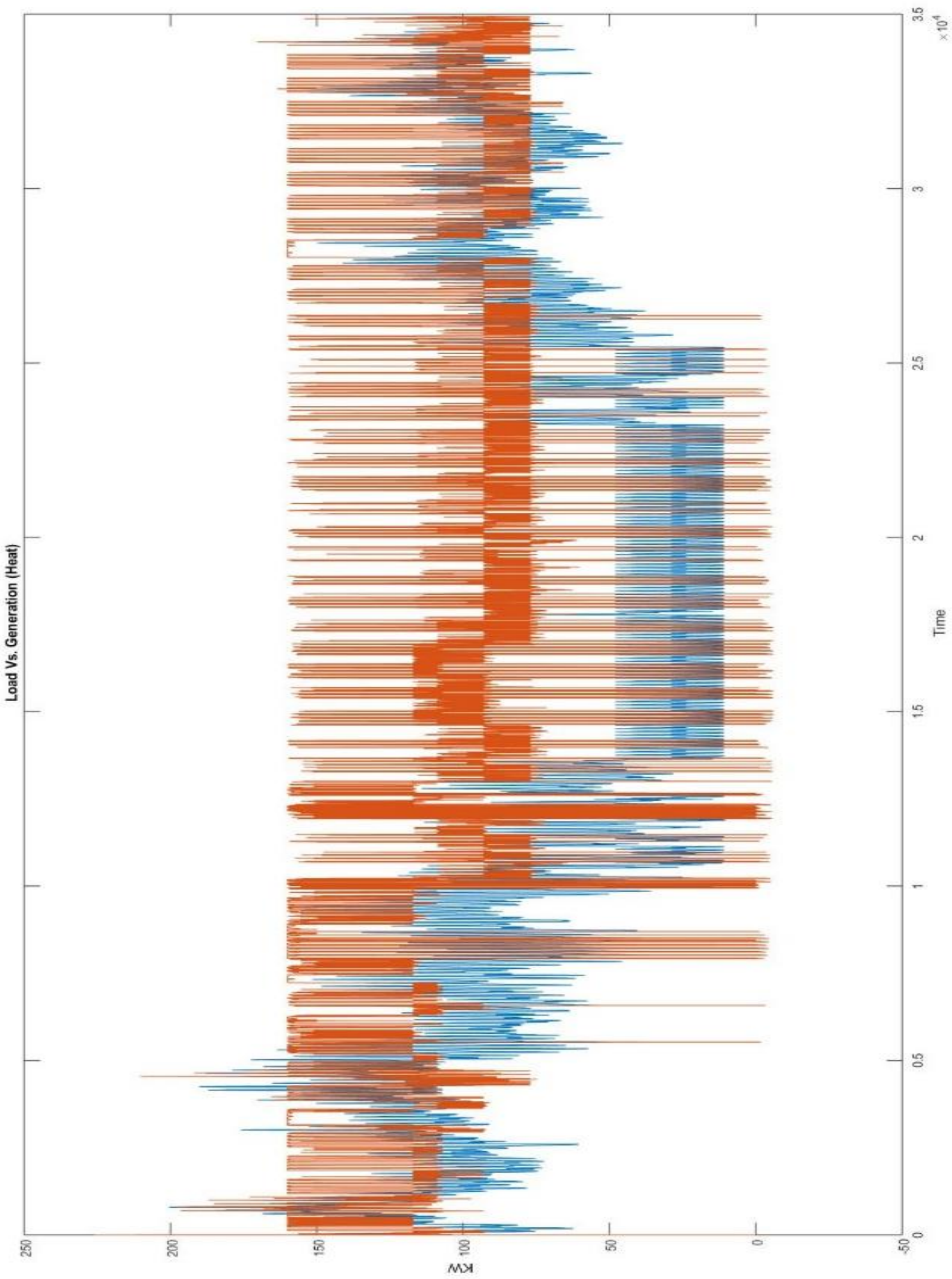


Figure V-11: Thermal Load vs. Generation - 2014

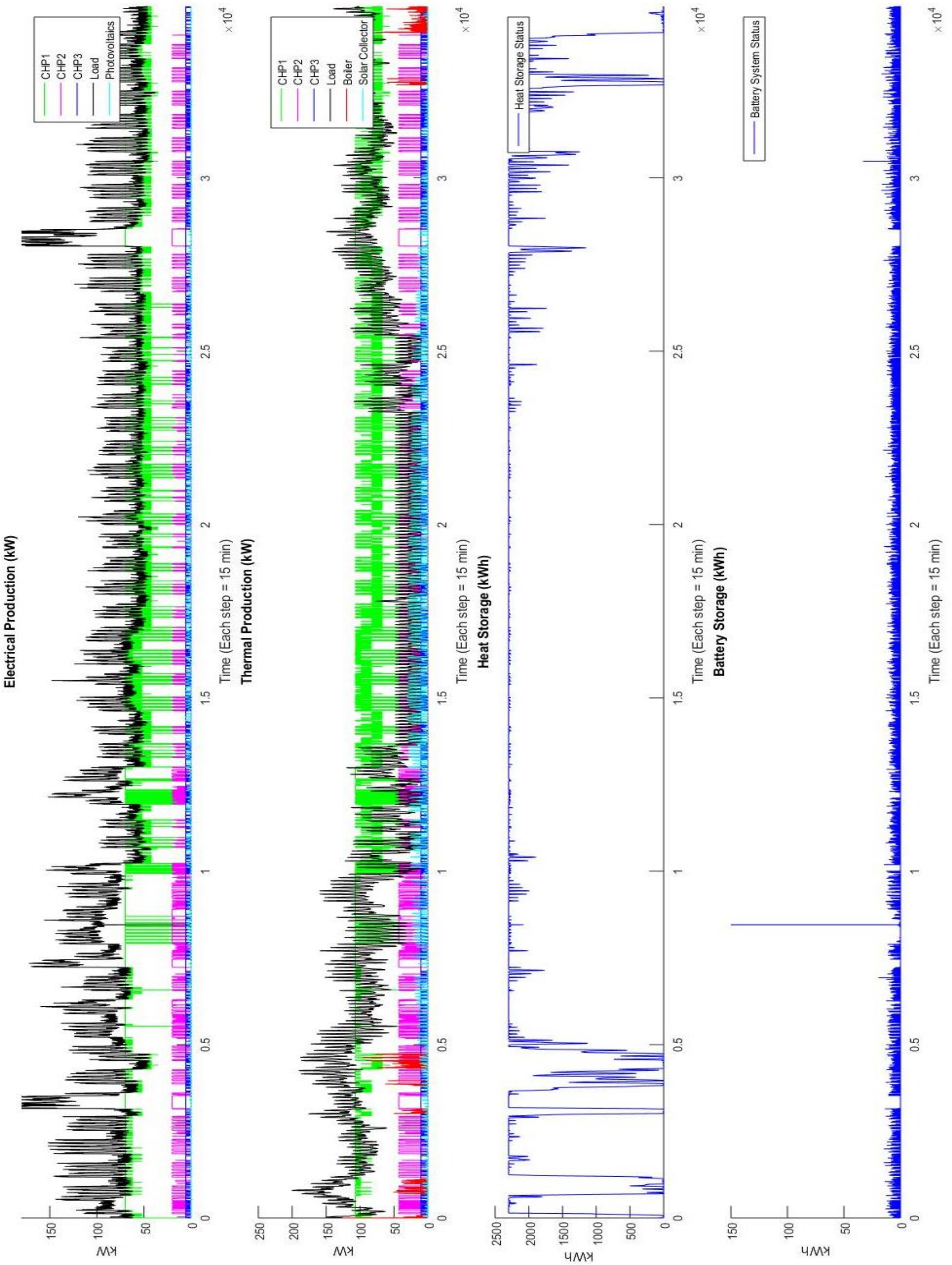


Figure V-12: Electrical/ Thermal production of each unit - Battery/Heat storage level -2014

3.4.2. Sample Cold Week / Day

It is possible to zoom in the results' diagram to observe the production/load in detail. Here the results for the sample cold week of 05.12.2014 to 13.12.2014 and the sample day of 10.12.2014 is presented.

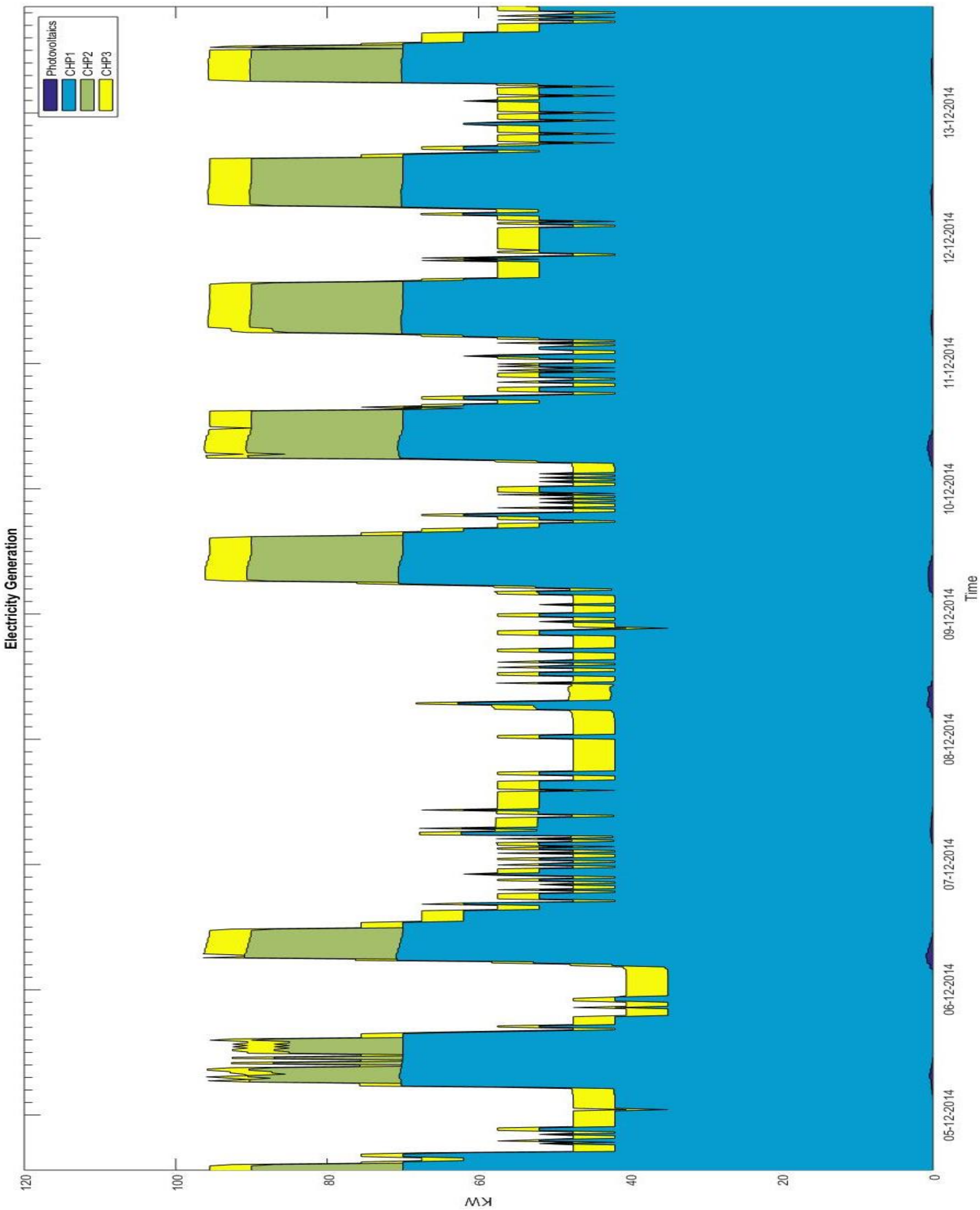


Figure V-13: Electricity generation- 05.12.2014-13.12.2014

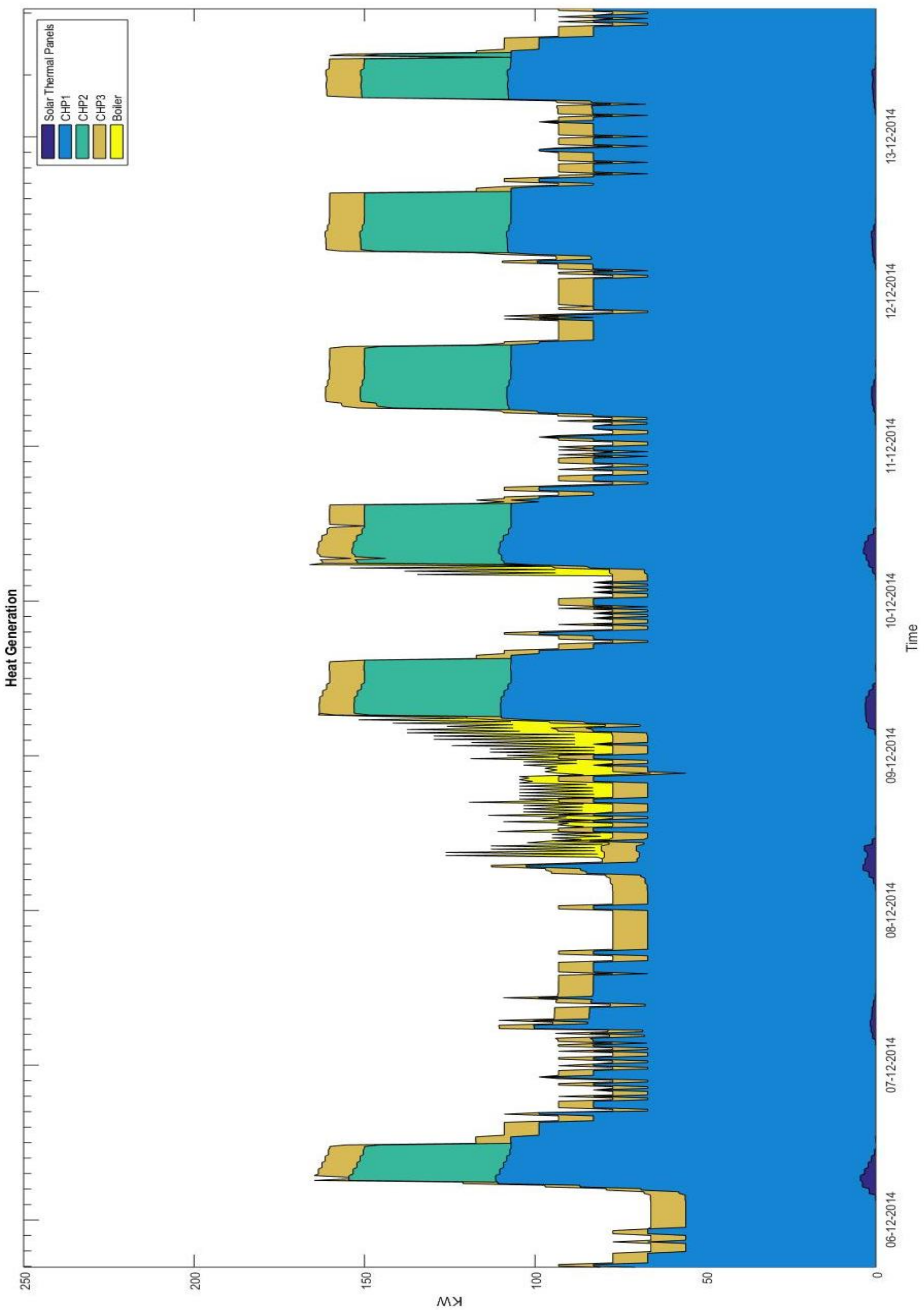


Figure V-14: Heat generation- 05.12.2014-13.12.2014

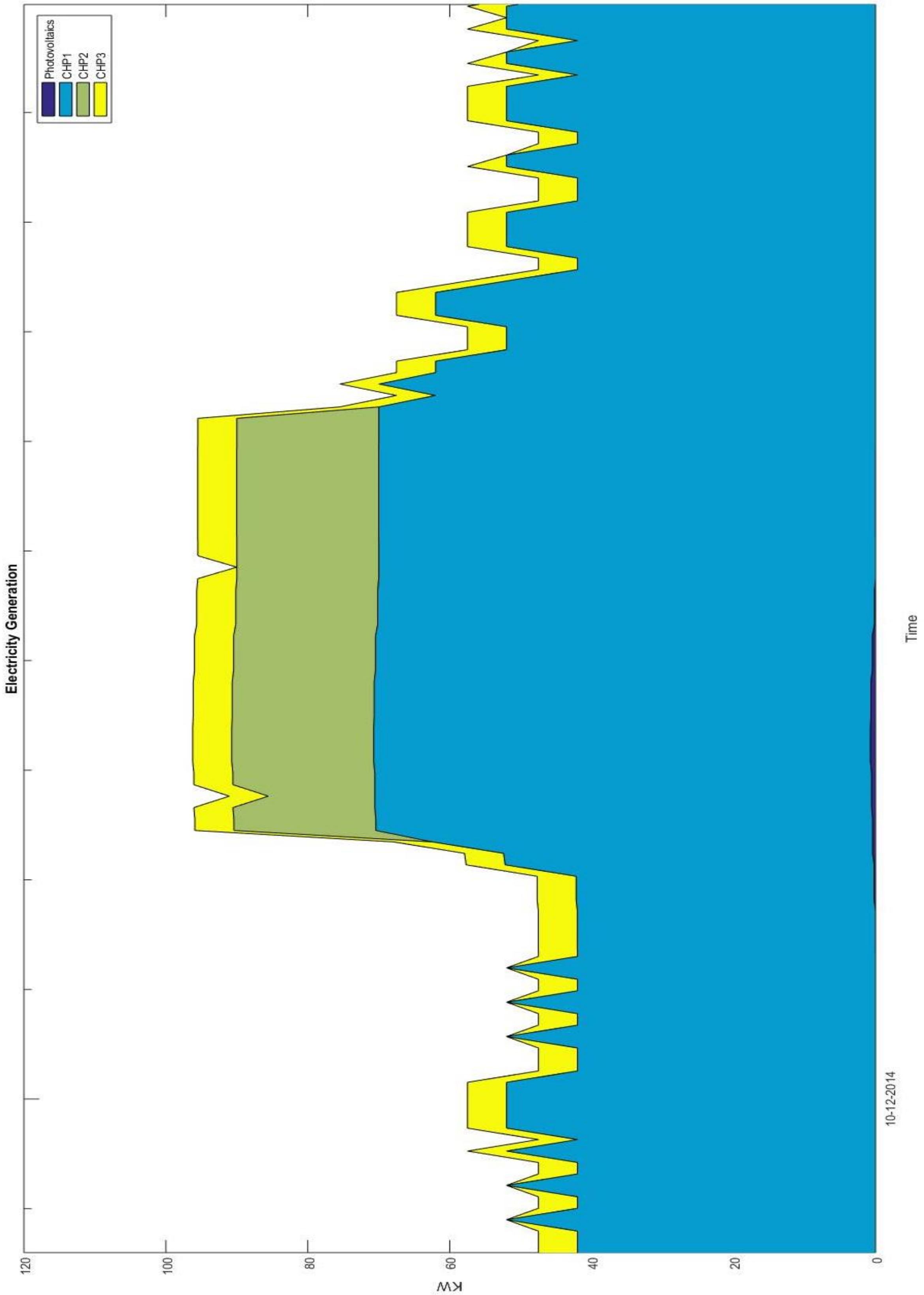


Figure V-15: Electricity generation- 09.12.2014

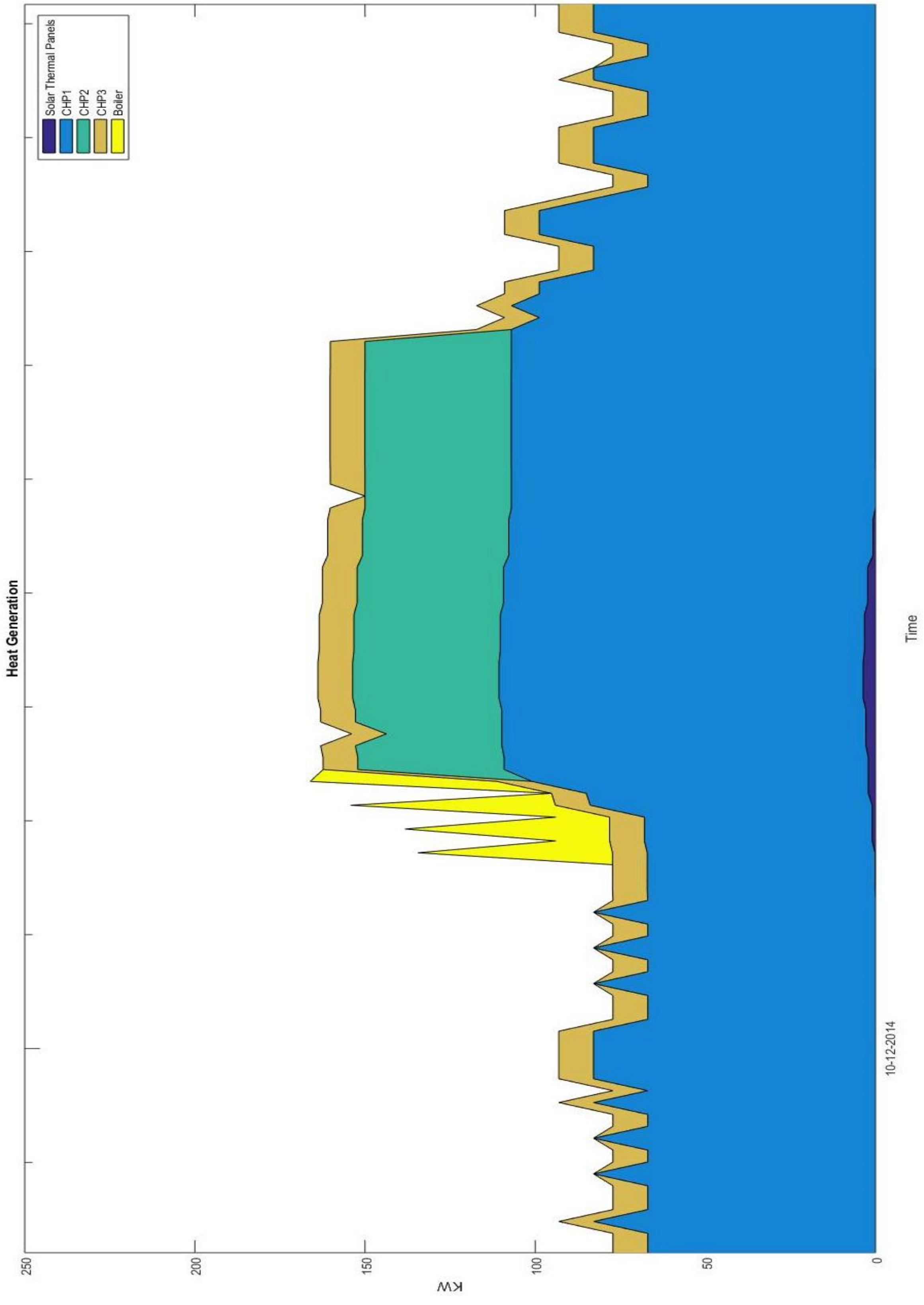


Figure V-16: Heat generation- 12.12.2014

3.4.3. Sample Hot Week / Day

The results for a sample hot week from 09.06.2014 to 16.06.2016 and also a sample day of 15.06.2014 is shown in the figures V-17 to V-20.

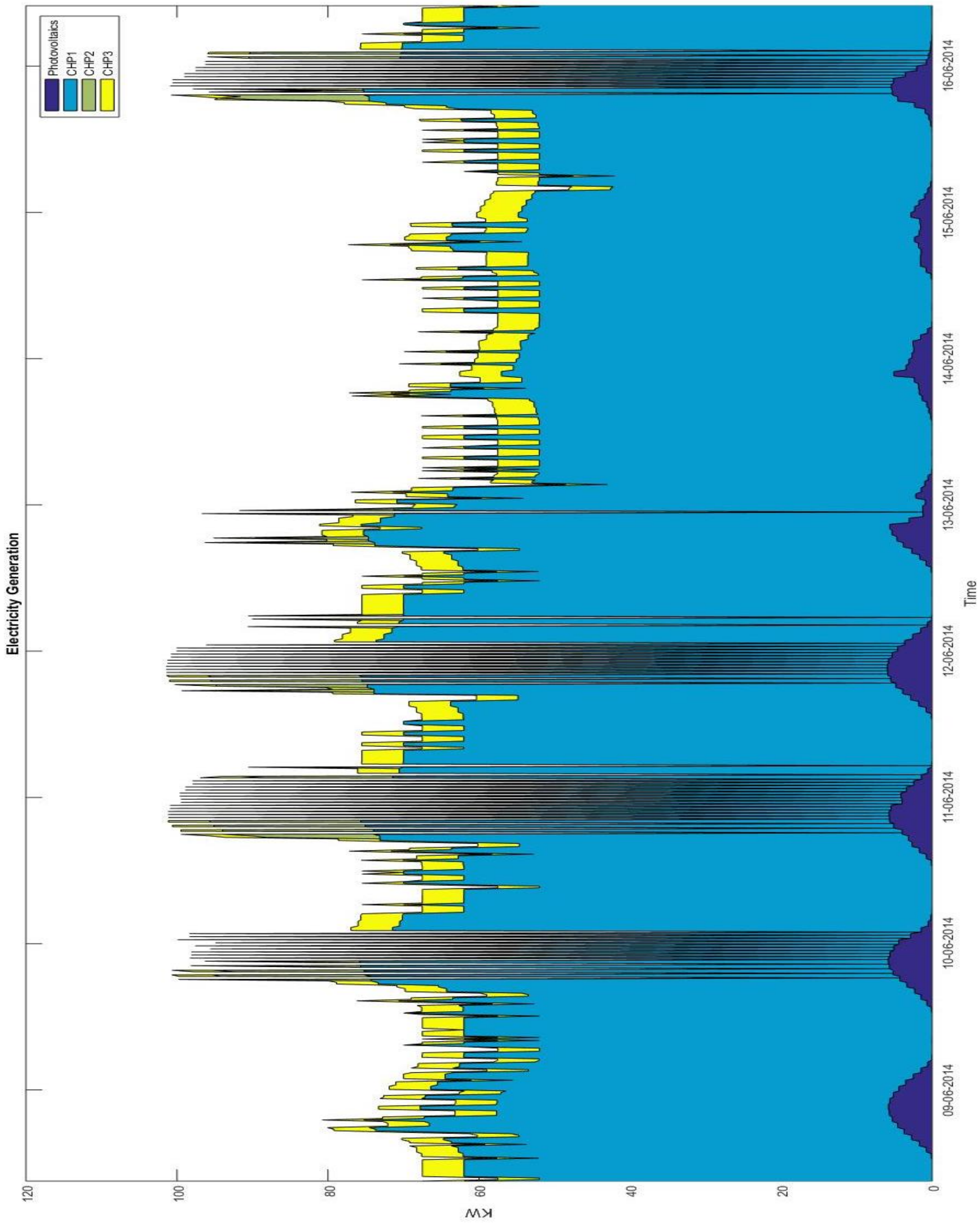


Figure V-17: Electricity generation- 09.06.2014-16.06.2014

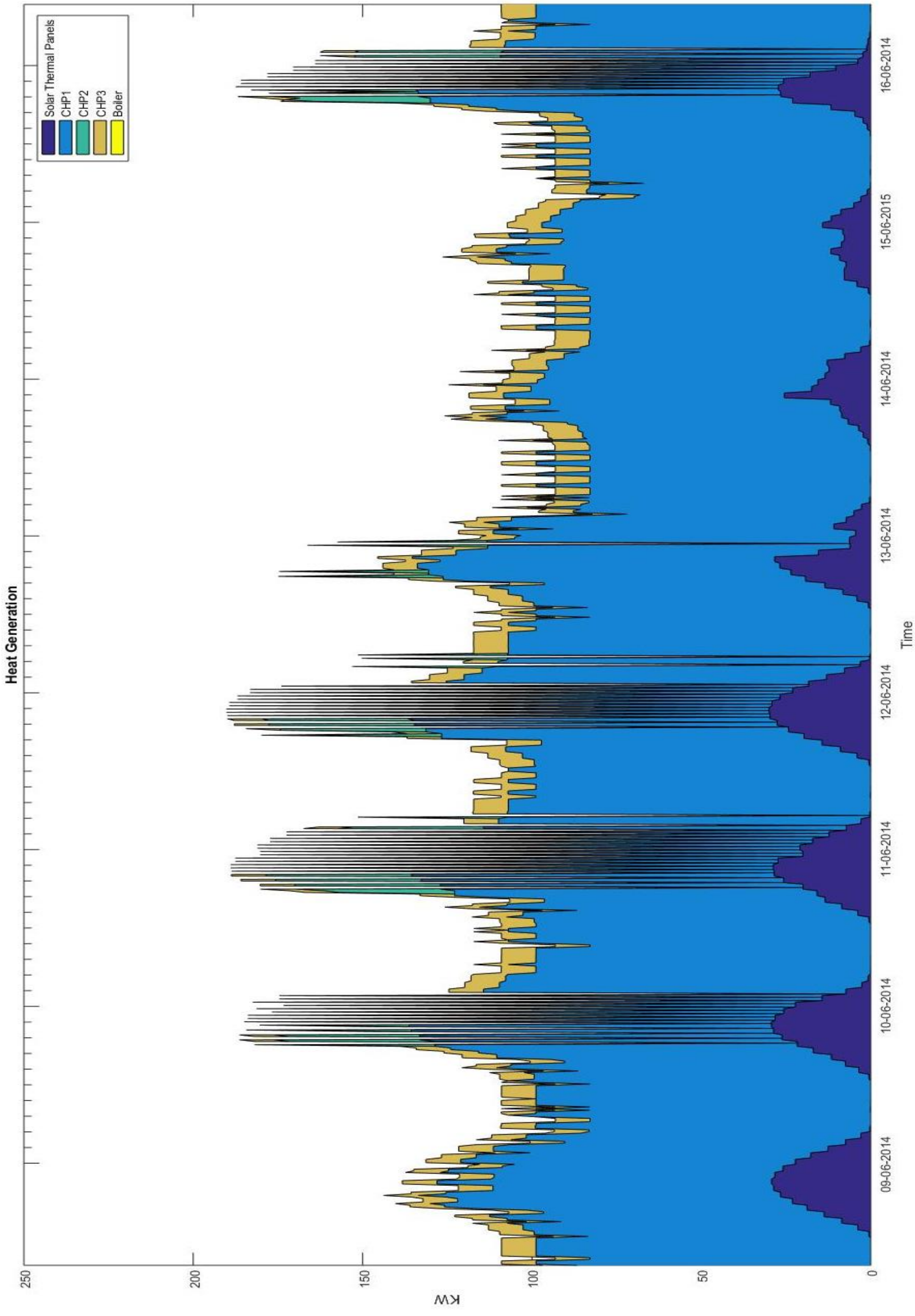


Figure V-18: Heat generation- 09.06.2014-16.06.2014

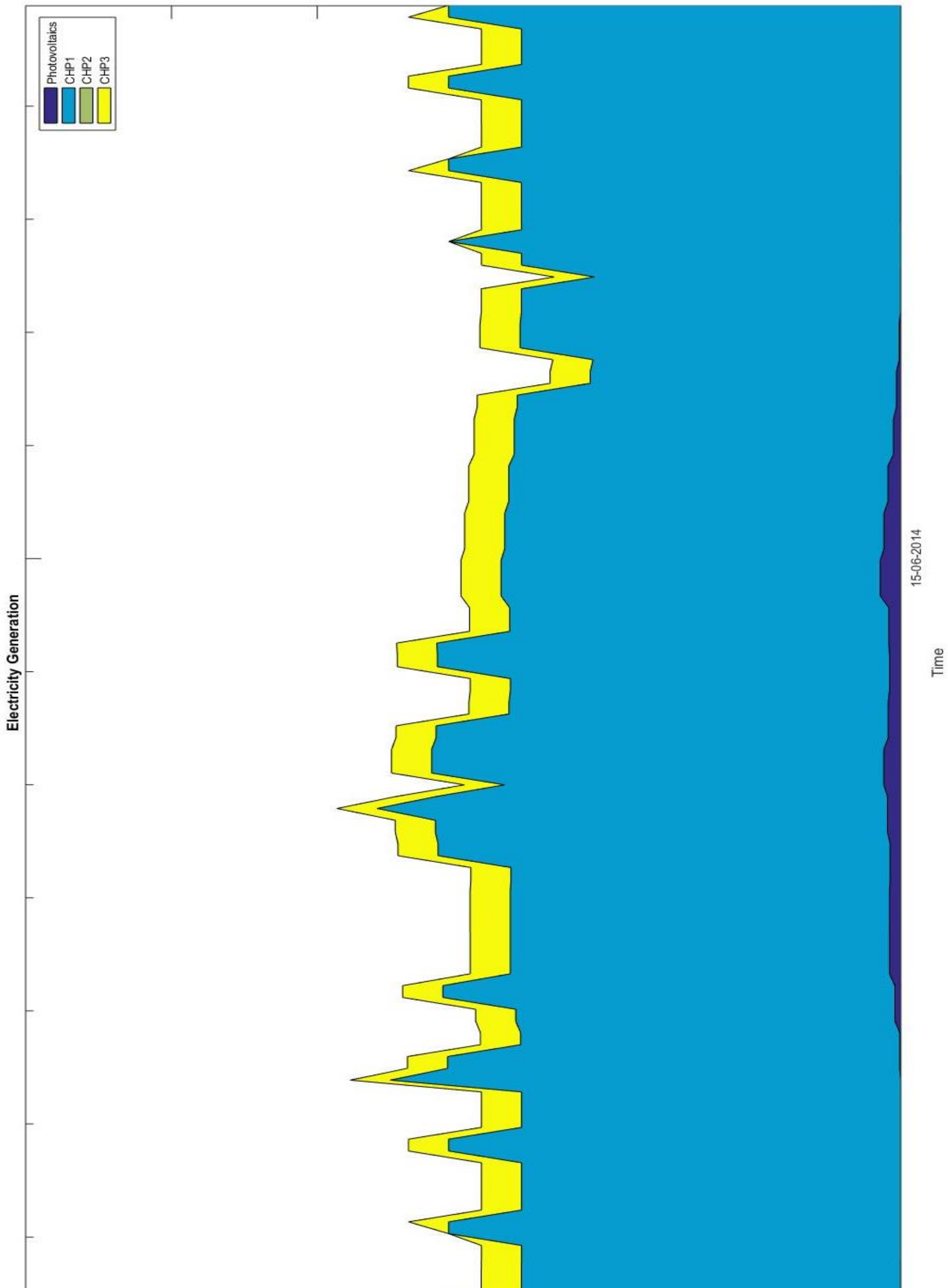


Figure V-19: Electricity generation- 15.06.2014

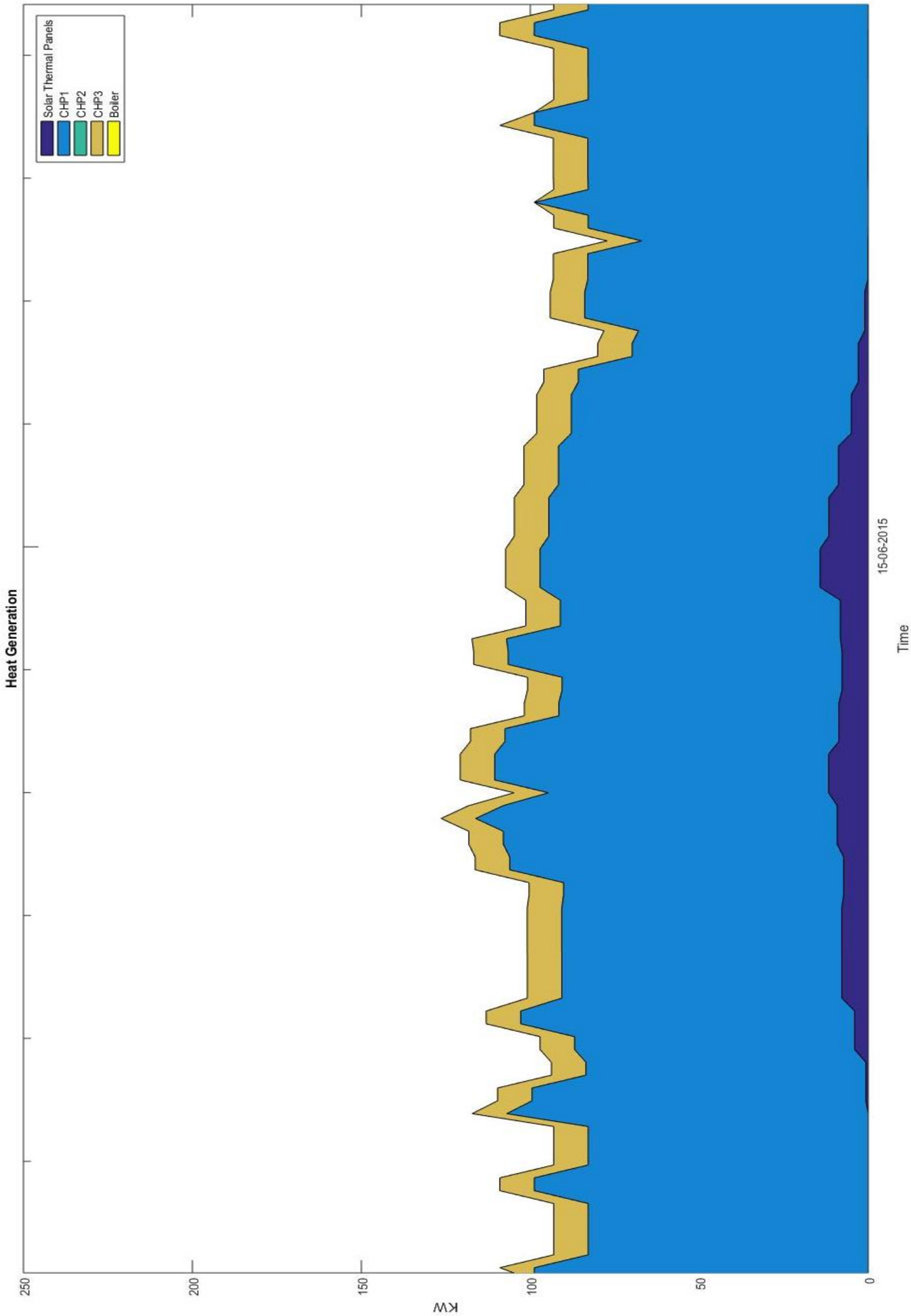


Figure V-20: Heat generation- 15.06.2014

3.5. Model Verification

The results presented in the part 3.4. were all in accordance with the real recorded data from 2014 by the controlling room in CUTEC institute. However, since some of the units such as the Buderus CHP unit were added later than 2014 the implemented model could not be completely verified by the actual historical measured data in the CUTEC energy park.

To make a better verification of this model, the ***EnergyPro software*** was used. Energy Pro is a relative complete modelling software package. It can model all types of electrical and thermal generation including all renewable units. Also, there are different blocks for other energy facilities such as energy storages. It is possible to choose between strategies for optimization of the whole system. The user can also define the time intervals for the calculation in the simulation system.

All the units in the energy park were modeled in the EnergyPro software and all the features –as far as possible- were set to be exactly similar to the ones implemented in MATLAB. (Figure V-21) The final results, percentage of production and the hours of application of each unit were all in accordance with the results from the MATLAB implementation with a neglectable difference, which are presented in Table V-2. These difference were the results of different efficiency rates for the CHP units, because the efficiency of CHP units cannot be set in the EnergyPro software.

Table V-2: Production rate and percentage of different units- EnergyPro Simulation

Unit	Electricity Production (MWh)	Percentage of total electricity production	Heat Production (MWh)	Percentage of total heat production
Boiler (Buderus)	-	-	12	1,1 %
CHP 1 (Buderus)	546,9	85,9 %	820,4	78,2 %
CHP 2 (Powertherm)	49	7,7 %	117,5	11,2 %
CHP 3 (Senertec Dachs)	30,8	4,8 %	62,4	6 %
Solar Collectors	-	-	37,3	3,6 %
Photovoltaics	9,8	1,5 %	-	-

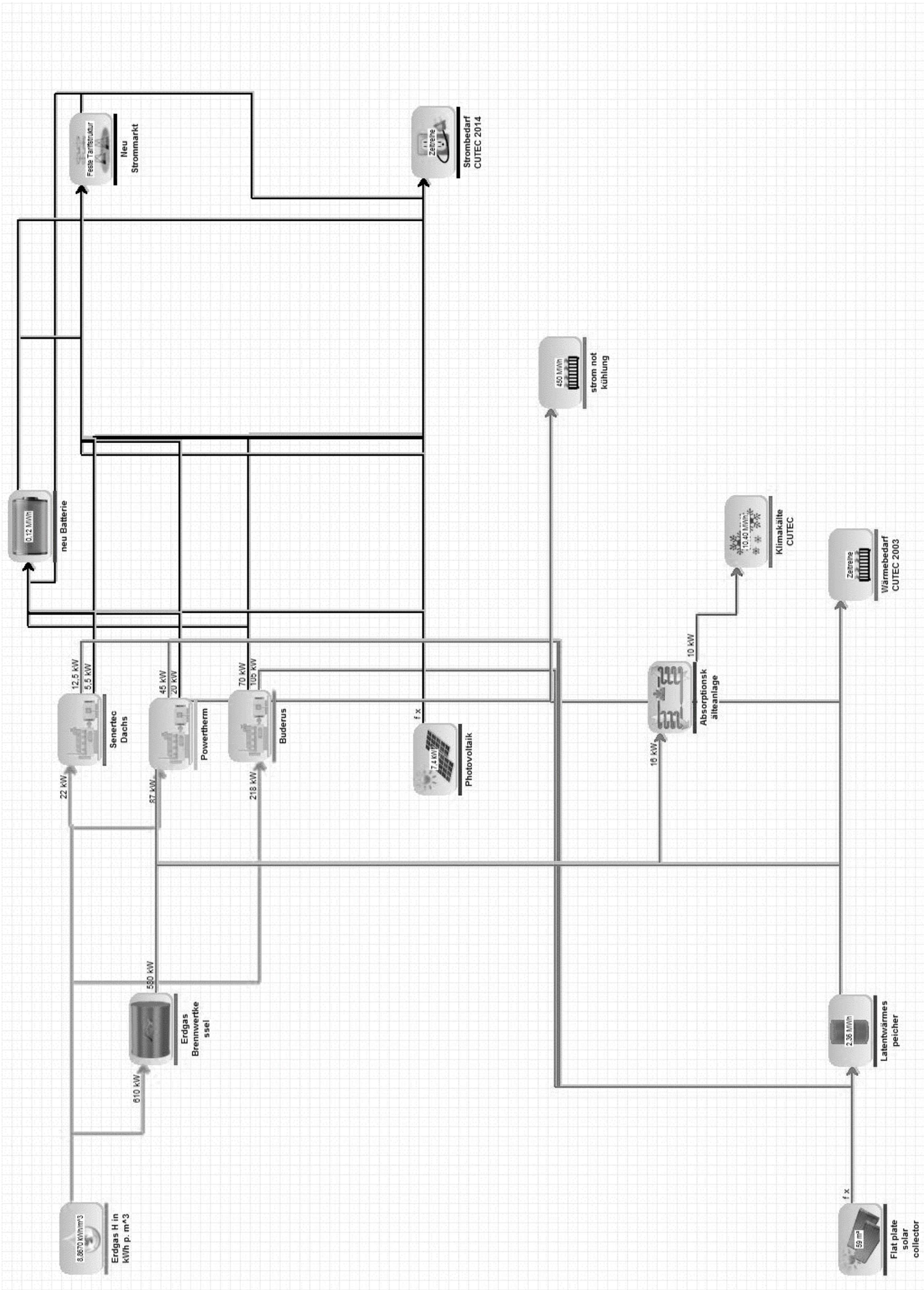


Figure V-21: Energy Park model in EnergyPro

VI. Chapter 6

1. Energy System Structure in Iran

Energy is one of the most important fields in economic activities and studies. The national security of many countries in the world depends on a secure access to energy resources and reliable energy supply. Therefore, the energy supply plans and optimal consumption play an important role in governmental master plans. Iran is known as a country with lots of hydrocarbons reserves in the world. Nearly 11 percent of global oil reserves (fourth largest) and 16 percent of global gas reserves (second largest) are located in Iran. Renewable energy resources are also widely available in Iran with high potentials. (A. kazemi, 2014) Iran is the second largest economy in the Middle East and North Africa region (MENA) after Saudi Arabia with an estimated Gross Domestic Product (GDP) of 406.3 billion dollars.

Considering the limitations related to the increase of crude oil and natural gas production, the growing consumption of oil and gas products, the dependence of Iran's economy and the general budget on the revenues from the sale of crude oil and also the high rates of natural resources exploitation and environmental problems in Iran, the need for an optimization in the supply and demand sectors of energy is an undeniable fact. In the next part, the current energy system in Iran will be introduced and the relevant data on energy production and consumption will be provided. The future development plans of the renewable energies in Iran will then be discussed and finally the proposed model for a load forecast and decentralized energy system analysis in this dissertation will be analyzed to study the transferability of them into Iran's energy system.

2.1. Current Situation

Primary energy supply in Iran includes the amount of generated and imported energy carriers after deduction of exports, supplies to the international ships, gases and other collectible energy including storage or withdrawal from storages as primary energy for domestic usage. Table VI-1 shows the share of different energy carriers in primary energy supply in Iran in year 2014:

Table VI-1: Share of different energy carriers in primary energy supply in Iran in 2014

Oil	Natural Gas	Coal	Nuclear Energy	Renewables
52.14 %	46.62 %	0.20 %	0.33 %	0.38 %

2.1.1. Oil

Iran's oil fields consist of reservoirs and oil basins which are mostly located in the southern part of Iran. Since Iran shares many fields with Iraq, Kuwait, Saudi Arabia, Qatar, United Arab Emirates and Oman, it has focused its executive and extraction activities on these fields and their development. The total recoverable reserves of crude oil and condensates were 358 billion barrels in 2014. With this share of crude oil and conventional gas condensates, Iran is ranked fourth in the world after Venezuela, Saudi Arabia and Canada.

2.1.2. Natural Gas

Based on the report from ministry of energy, Iran has had 23 active on- and off-shore natural gas fields in 2014. The total amount of recoverable natural gas reserves were estimated about 33.44 trillion cubic meters in 2014, which made Iran to have the second largest natural gas reservoir in the world after Russia.

2.1.3. Coal

Iran is in a very special condition in case of coal resources. The main part of the Iran's coal reserves had been formed during the Mesozoic middle ages period. The formed coal is very thin and located very deep in earth. Iran's coal layers are between 30 to 150 centimeters and located at a depth of 100 to 300 meters, which has a loose soil coating that makes the extraction process very difficult. The extraction rate of coal from 111 active mines in Iran was almost 3341.2 thousand tons in 2014.

2.1.4. Renewables

Iran's potential for renewable energy is huge by possessing some of the best combined wind and solar resources in the region. The nominal power of renewable power plants (excluding the hydro power energy) has been tripled since 2006 and reached 168.6 MW in year 2014. The share of electricity generation from wind, solar and biogas power plants were, however, roughly 0.2 percent in this year, which shows the huge dependency of country on the gas, thermal and combined cycle power plants. As mentioned, Iran has a great potential in renewable resources. Based on a pilot study in 2013 led by the Ministry of Energy, there is a potential of almost 130 GW only from wind energy in the whole country. Also, having more than 300 sunny days in almost 66 percent of the country's area with an average of 4.5 to 5.5 kWh/m^2 radiation, makes Iran a country with one of the highest potential in solar energy. Also in geothermal energy, a potential of about 200 MW has been estimated. VI-2 shows the operational renewable power plants together with the under construction and planned projects.

Table VI-2: Renewable energy power plants (Operational and potential) (Green, 2017)

		Wind (MW)	**Solar (MW)**	**Biomass (MW)**	**Small Hybrid (MW)**	**Marine (Waves) (MW)**	**Total (MW)**
Operational		53.88	0.514	10.56	0.44	-	365.39
PPA concluded	ART. 133	551.5	10.3	-	0.17	-	561.97
	ART.19	959	5	3	3.6	-	967.6
Construction Permit or preliminary issued		5662.96	1615.02	194.29	23.5	0.15	7495.92
Cancelled permit		6670	1281	103.9	35.31	-	8090.21
Total Capacity		13897.34	2911.83	308.75	63.02	0.15	17181.09

2.1.5. Electricity

The current total installed power capacity of Iran is 73 GW. The electricity consumption has grown steadily in Iran during the last decade with an average of 7 percent annually. In 2014, the whole country's electricity demand was covered by the power plants affiliated with the ministry of Energy, large industries and private sectors[6], consisting of 24 thermal power stations, 65 gas power plants, 24 CHP and distributed generation units, 19 combined cycle power plants, 46 diesel power stations, 50 hydro-Electric power plants (large, small, micro and pico units), 224 wind turbines, 4 photovoltaic units and 6 bio gas power plants. With the installation of new power plants (2.9 GW), the total nominal electricity generation capacity of the country reached 73.2 GW in 2014. The nominal power capacity of these power plants are listed in the table VI-3. In this year, the total electricity generation of power plants in Iran was roughly 274.4 TWh, which showed an increase of 4.6 percent compared to the year before. This amount was provided by power plants affiliated to Ministry of Energy (43.2%), Private sector (52.9%), large industries (2.3%) and nuclear energy organization (1.6%). The total export of electricity in this year was roughly 9659.9 GWh, where the total import was 3771.5 GWh.

Figure VI-1 shows the share of different types of power plants in the total electricity generation in Iran in 2014:

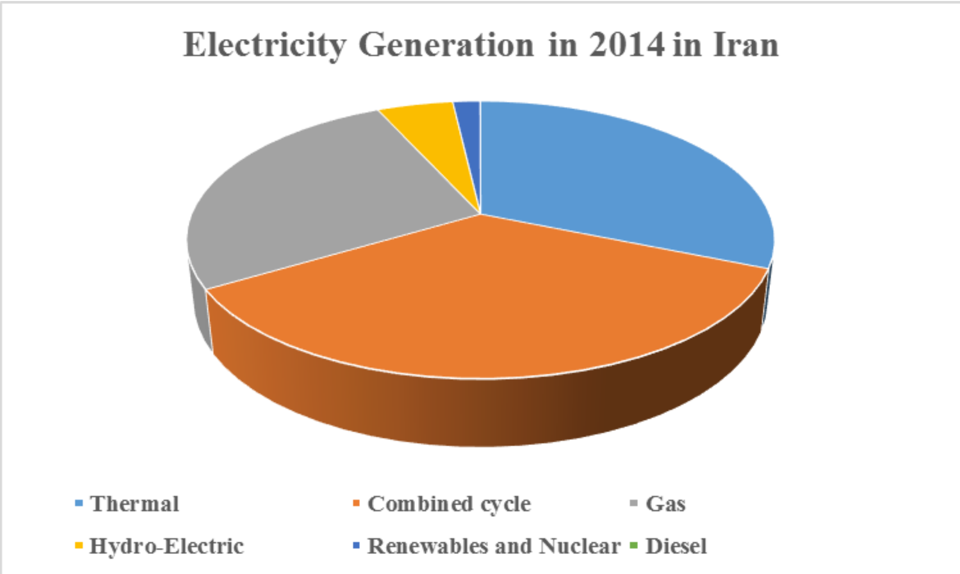


Figure VI-1: Share of different types of power plants in the total electricity generation in Iran in 2014

[6] Prior to the part-privatization in 2004, the electricity market was a complete state monopoly held by TAVANIR. Since 2004 the government has tried to increase the private participation in the market.

Table VI-3: Nominal power capacity of different power plants

	thermal (MW)	Gas (MW)	Combined Cycle (MW)	Diesel (MW)	Hydro Electric (MW)	Wind (MW)	Solar (MW)	Total (MW)
Ministry of Energy	11241.1	7195.2	4274.6	439.4	10788.9	98.9	0.07	34038.1
Industries	588.6	4992	-	-	-	-	-	5580.6
Private sectors	3399.5	14224.8	14218.5	-	-	54.6	0.5	32512*

** The private sectors include the Biogas power plants*

Before introducing the future development plans of Iran in Energy sector, it is worth to review the share of final consumers categorized by each energy carrier. Table VI-4 gives an overview about the final energy consumption percentage in different sectors:

Table VI-4: Final energy consumption percentage in different sectors categorized by each energy carrier

	Residential, Public, Commercial	Industry	Transportation	Agriculture	Non-Energy Usage
Oil products	9.39 %	7.38 %	61.48 %	4.37 %	17.38 %
Natural Gas	47.03 %	34.73 %	6.39 %	1.22 %	10.65 %
Coal	2.28 %	28.35 %	-	-	69.37 %
Renewables	100 %	-	-	-	-
Electricity	48.03 %	34.18 %	0.16 %	15.89 %	1.73 %

2.2. Future Development Plans

The Iranian government and policy makers have shown a significant interest in the renewables in recent years. This is mainly due to the domestic power demand increase, subsidies omitting[7] and the need for a more secure energy system. Following this interest, they have set an ambitious target of 5 GW of nominal capacity from the renewable power plants by 2020. Also, Iran's renewable organization, SUNA, has set very attractive feed-in tariff with the guarantee of purchase for 20 years. (Ghobadian, 2010)

Beside these, there have been many scientific and research activities in the field of renewable energy in Iran in recent years, which will be realized later in form of more practical processes such as establishing power plants.

Based on recent statistics, there will be huge investments in wind energy (with almost 25 MW operational power in near future) (Iran Energy Balance in 2010) . There have been many efforts to absorb the attention of private sectors. Based on the 4th Socioeconomic and Cultural Development Plan (2005–2010), a share of at least 270 MW is expected from the private sector which will be mainly in wind energy. Based on this major plan, the government must prepare the basic infrastructures for the other renewables such as biomass and solar power plants. In this way, more private sector participation can be expected in these fields too.

Currently a 250 kW geothermal power plant is under construction in Shiraz and two more units with 5 and 50 MW capacities will follow. There are also two upcoming projects which include a solar power plant with a nominal capacity of 17 MW and a biomass plant with 10 MW capacity.

There are also some ongoing projects funded by the global environment facility (GEF) that provide a number of international and national consultant missions and studies. Following these studies, a project will be started in Gilan province (Manjil) with the aim of constructing a wind park with 25 MW nominal capacity. This project will be a part of the national development plans aiming a 100 MW of wind-powered energy.

[7] Iran was the world's highest provider of fuel subsidies in 2009 with almost 10% of GDP equal to 40 billion dollars.

2.3. Transferability of Proposed Model to Iran's System

As mentioned in the last parts, Iran has a great potential in renewable resources. Having 130 GW of potential capacity in wind energy, being located on several major wind corridors and consistent, high wind years in the North East and North West, make the wind energy one of the interesting and reliable option for a decentralized energy system. On the other hand, having 300 clear sunny days with solar radiation between 4.5 to 5.5 kWh/m^2 in a total area of 1600,000 Km^2 allows two third of the country to provide the energy locally. Furthermore, 15 potential sites for solar power plants with an average capacity of 200 MW each, are located and the first pilot projects are running. This variety shows that there is a great potential in Iran to generate the energy locally in form of distributed energy systems. (Iran Energy Balance in 2012)

Due to the fluctuating generation of energy in the systems based on renewable resources and the complexity and redundancy of such systems, especially when they are designed to operate as island energy systems and also integrating the new ideas for heat storage or power to X units, it is necessary to have an energy system model to perform the optimization procedure and other experimental studies prior to real construction.

The energy system model presented in this dissertation is designed to be flexible and include all the common units for a decentralized energy system. This model can provide the user with the information about the technical operation of the whole system. It also provides the generation statistics and the detailed operation plan of all units.

Considering the national energy developments in Iran, which include major infrastructure construction for micro grids and distributed energy systems to provide the required energy locally, the presented model can be very advantageous for the future energy system planning.

This model can also be helpful for a potential site for a decentralized energy system based on solar energy. Provided that the electrical and thermal load and also the solar irradiation of the site (or a similar site) is known, the system can be simulated with this model to find the optimal combination of the units and nominal capacity of them. Also it is possible to predict the future load based on the method introduced in this dissertation to achieve a better understanding of the future behavior in the system, so that the energy system will be designed in a more optimal way. This will reduce the operation costs in future.

Needless to say, this model can be developed to cover the potential generation from wind, geothermal energy and etc. which make the model applicable also for the sites with the potential for the combination of these resources.

VII. Chapter 7

1. Conclusion, Final remarks

The importance of a precise load forecasting in energy system designing has been known since many years ago. To achieve an efficient, secure and economically optimal operation in energy systems, it is necessary to analyze the power system operation in time and perform fast reactions to changes in electrical load. These are highly dependent on an accurate short-term load forecasting. On the other hand the future development plans and the energy demand management such as peak load reduction need sufficient knowledge on the future electrical load in longer time horizons.

This dissertation has tried to propose a new approach toward electrical load forecasting based on time series analysis together with concept drift and chaos theories. The basics of this methodology were introduced in chapter 2 by explaining the idea of concept drift. The electrical load data from CUTEC institute in the year 2014 were chosen as the case study. This data was then analyzed by mutual information function to find the similarities and repetitions. This was inspired by the idea of concept drift. The data was then categorized with the possible lags achieved by this function. This process was done for each time horizon separately based on trial and process in the final step. After that the chaos theory was discussed and the existence of it in the sample time series was inspected. A chaotic time series is based on a deterministic system and it can be concluded that it has enough information in it to be predicted by the proposed method in this dissertation. The chaos existence was examined based on three different methods, which were explained in detail in chapter 2. Chapter 4 introduced the ANFIS. This system has the advantages of both Artificial Neural Networks in self-training and manipulability of fuzzy logic. The methods, which are solely based on ANN, process the data through nonlinear activation functions, but they have a block box nature and the process lacks controllability. This problem has been solved in ANFIS by using the characteristics of Fuzzy systems. An ANFIS-based model was developed in MATLAB to predict the electrical load. The categorized data based on the best lags achieved by mutual information function was then fed into this model systematically and the results of weekly, monthly and seasonal electrical load were presented. The results were then compared with real data from CUTEC in the same year and the important indices in load forecasting studies such as MAPE were given, which were significantly lower than most reported similar cases in studies with almost the same datasets and prediction horizon, confirming that the proposed method shows a more efficient, time-saving, and more accurate approach to the problem of electrical load prediction.

The proposed method in this dissertation shows a more efficient, time saving and accurate approach toward the problem of electrical load forecasting.

In chapter 5 a simulation model for a decentralized energy system was introduced. This model is implemented in MATLAB and SIMULINK. It contains the conventional units for a distributed energy generation system, which can also work in island mode. The user can feed the predicted electrical and thermal load through the designed GUI as inputs, modify the priority of the units and also add extra details such as the nominal power, efficiency or capacity of storages. The model provides the information on hourly operation of each module in form of excel data and several graphical illustrations and other overall information.

This model was validated with the Energy Park in CUTEC institute and the sample results for hot and cold week/day were presented. The overall results for the sample year of 2014 were then compared with the results from a commercial energy management software (EnergyPro). The results were in the same scale with neglectable difference of less than 1%.

2. Future works

The proposed method in this research analyzes the electrical load data as time series and performs the prediction solely based on it with no other elements involved. The author believes that the time series already contents the important information about the parameters influencing the load. These information are just needed to be extracted and used for the proper training process. Therefore, this method can be very helpful for the systems with little information on meteorological data or other influencing factors. Furthermore, this approach can also be applied on similar data, which come in forms of time series such as thermal load data, solar irradiations in a specific area or even market prices. Needless to say, for such time series, the existence of chaos should be investigated in the first step, since it is a crucial factor in this method that verifies the predictability of the sample time series.

This model was trained by a sample data of one year. The author believes that using a longer time period and analyzing such time series help the training process and the results can improve. It will also be possible to forecast the electrical load for a longer period. However, for a longer time horizon in prediction, it should be considered that there are some important factors which have crucial influence on the predicted electrical load curve. (Quilumba, Lee, Huang, Wang, & Szabados, 2015) Therefore, although the load forecasting based on the historical data may help to have good understanding about the future load, it cannot completely predict the precise load curve deformation, especially for longer time intervals. To achieve more accurate results, some correction and supplementary factors related to new technologies should be taken into consideration such as penetration rate of electrical vehicles and their charging loads, smart management of end-user consumption, the usage of surplus electricity for new concepts in system like power to X, etc. (Chen, 2016) (Kavousi-Fard, 2016).

One of the other practical future development of the method introduced in this dissertation is the implementation of a real-time predictor in such a way that the newest load data will be fed in the training process and be used to retrain the system. It is also possible to create a monitoring system which gets the current load data and the already forecasted load as input and visualize the difference in real time.

The implemented Energy Park model can also be improved by adding other units of distributed energy generation such as wind turbines or power to X units to model a poly-generation energy system. It would be also helpful to add more details to some units in this model. For instance to add the details of charging and discharging processes in batteries or to take the heat losses of building into consideration.

VIII. References

A Khosravi, S. N. (2010). Construction of Optimal Prediction Intervals for Load Forecasting Problems. *IEEE Transactions on Power systems, vol. 25*, pp. 1496-1503.

A. kazemi, H. S. (2014, September). A review on energy system models in Iran. *Energy Policy and planning* , pp. 5-28.

Abarbanel, H. D. (1996). Analysis of observed chaotic data. *Physics Today 49. Springer*, pp. 25-37.

Abdullah S. Al-Fuhaid, M. A.-S. (1997). Neuro-short-term load forecast of the power system in kuwait. *Applied Mathematical Modelling 21 (4), Elsevier*, pp. 215-219.

Administration, U. E. (2014). *Renewable Energy information 2014.* http://www.eia.gov/.

Alfares, H. K. (2002). Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science 33.1*, pp. 23-34.

Amit Jain, E. R. (2009). Short term Load forecasting using Fuzzy Adaptive inference and. Nature Biologically Inspired Computing (*NaBIC).*

Anderson, J., & Rosenfeld, E. (1988). Neurocomputing: Foundations of Research. *MIT Press, Cambridge*.

Andrew Ng, J. N. (2016). *Deep Learning Tutorial.* Retrieved from http://deeplearning.stanford.edu/tutorial/

Bakirtzis A. G. et al. (1995). Short Term Load Forecasting using Fuzzy Neural Networks. *IEEE Transactions Power*, pp. 1518 - 1524.

Bayram Akdemir, N. Ç. (2011). Long-term load forecasting based on adaptive neural fuzzy. *2011 2nd International Conference on Advances in Energy Engineering (ICAEE 2011)*, pp. 794-799.

Bishop, R. (2017). *Chaos.* The Stanford Encyclopedia of Philosophy, *Online articles.*

Bri-Mathias S. Hodge, S. H. (2011). A multi-paradigm modeling framework for energy systems simulation and. *Computers and Chemical Engineering 35, Elsevier* , pp. 1725-1737.

Buzug, T. h., & Tom, P. G. (2008). Optimal Delay time and Embedding Dimension for Delay-Time Coordinates by Analysis of the Global Static and and Local Dynamical Behaviour of Strange Attractors. *Phys. Rev. A, 45*, pp. 7073-7084.

Cao, L. (1997). Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena 110.1*, pp. 43-50.

Cencini, M., & al, e. (2010). Chaos From Simple models to complex systems. *World Scientific, ed*., pp. 111-124 , 228-234.

Chen, Y. H. (2016). Electric Load Forecasting Based on a Least Squares Support Vector Machine with Fuzzy Time Series and Global Harmony Search Algorithm. *Energies 9.2*, pp. 70-72.

Chi Xie, Z. C.-1. (2006). Sequence Outlier Detection Based on Chaos Theory and Its Application on Stock Market. *Fuzzy Systems and Knowledge Discovery*, pp. 1221-1228.

Chicco, G., Napoli, R., & Piglione, F. (2001). Load pattern clustering for short-term load forecasting of anomalous days. *IEEE Porto Power Tech Proceedings (Cat. No.01EX502).*

Çunkaş, H. H. (2015). Short-term load forecasting using fuzzy logic and ANFIS. *Springer London, Neural Computing and Applications*, pp. 1355–1367.

Danladi, A. e. (2016). Use of Fuzzy logic to investigate weather parameter impact on electrical load based on short term. *Nigerian Journal of Technology 35.3*, pp. 562-567.

Ding, N. e. (2016). Neural network-based model design for short-term load forecast in distribution systems. *IEEE Transactions on Power Systems 31.1*, pp. 72-81.

Ertel, W. (2016). Grundkurs Künstliche Intelligenz, eine praxisorientierte Einführung. *eBook ISBN 978-3-8348-9989-7, Springer verlag*, pp. 67-100.

Esa Aleksi Paaso, Y. L. (2013). Development of New Algorithms for Power System Short-Term Load Forecasting. *International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 02– Issue 02*.

Esener I, Y. T. (2013). Short-term load forecasting without meteorological data using AI based structures. *Turk Journal of Electral Engineerin and Computer Science*, pp. 370-380.

Fayyad, U. G.-S. (1996). From data mining to knowledge discovery in databases. *AI magazine 17.3*, pp. 37-42.

Fazil Kayteza, M. C. (2015). Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems Elsevier*, pp. 431-438.

Filik UB, G. O. (2011). Hourly forecasting of long term electric energy demand using novel mathematical models and neural networks. . *Int J Innov Comput Inf Control 7(6)* , pp. 3545–3557.

Florian Grupp, F. G. (2007). Simulink Grundlagen und Beispiele. Germany: *Oldenbourg-Verlag*.pp. 19-27 and 93-104.

Gabriela Grmanova, P. L. (2016). Incremental Ensemble Learning for Electricity Load Forecasting. *Acta Polytechnica Hungarica*, pp. 97-117.

Gama, J. I. (2013). A Survey on Concept Drift Adaption. *ACM Comput. Surv.1, Article 1*, pp. 1-4.

Gang Li, C.-t. C.-y. (2007). Short-Term Load Forecasting Using Support Vector Machine with SCE-UA Algorithm. *Third International Conference on Natural Computation (IEEE).*

Ghobadian, B. N. (2010). Future of renewable energies in Iran. *enewable and Sustainable Energy Reviews 13*, pp. 689-695.

Gómez-Expósito, A. A. (2016). Electric energy systems: analysis and operation. *ISBN-13: 978-1138724792 , CRC Press.* ,pp. 2-30.

Grassberger. (1994). An optimized box-assisted algorithm for fractal dimensions. *Physics Letters A, Elsevier, Volume 148, Issues 1–2*, pp. 63-68.

Grassberger, P. a. (1991). Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena 9.1*, pp. 189-208.

Green Power. (2017). *Renewable Energy Opprtunities in Iran.* Germany.

Hashemifarzad, A. (2014). *Simulation model of a multi-level inverter (Master-Thesis).* TU-Clausthal. pp. 40-52.

Heiko Hahn, S. M.-N. (2009). Electric load forecasting methods: tools for decision making. *European Journal of Operational Research 199*, pp. 902–907.

Heinemann, G. D. (1966). The relationship between summer weather and summer loads - a regression analysis. *IEEE Transactions on Power Apparatus and Systems*, pp. 1144 - 1154.

Hilborn, R. (2000). Chaos and nonlinear dynamics: an introduction for scientists and engineers. *oxford university press.*,pp. 3-116.

Holt, C. C. (1967). Forecasting Trends and Seasonals by Exponentially Weighted Moving Averages. *Office of Naval Research Memorandum*, pp. 5-10.

Honade, H. P., & J., S. (2015, May). ANFIS Based Short Term Load Forecasting. *International Journal of Current Engineering and Technology*, pp. 1878-1880.

Hong, W. (2013). Intelligent Energy Demand Forecasting. In S. *Lecture Notes in Energy 10.* , pp. 11-40.

Hong, W.-C. (2011). Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy 36.9*, pp. 5568-5578.

Hooshmand, R.-A. H. (2013). A hybrid intelligent algorithm based short-term load forecasting approach. *International Journal of Electrical Power & Energy Systems 45.1*, pp. 313-324.

Igor R. Krcmar, P. S. (2010). A Class of Neural Adaptive FIR Filters for. *10th Symposium on Neural Network Applications in Electrical Enginnering .*

Islam, E. B. (2011). Comparison of conventional and modern load forecasting techniques based on artificial intelligence and expert systems. *IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5*, pp. 504-513.

Islam, S. M.-A. (2001). Forecasting monthly electric load and energy for a fast growing utility using an artificial neural network. *Electric Power Systems Research 34.1*, pp. 1-9.

João Gama, P. M. (2014, April). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR), Volume 46 Issue 4,*, pp. 44.1 -44.37.

Junran Peng, S. G. (2017). Study of the Short-Term Electric Load Forecast Based on ANFIS. *32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE*, pp. 832-836.

Kavousi-Fard, A. A. (2016). A new fuzzy-based combined prediction interval for wind power forecasting. *IEEE Transactions on Power Systems 31.1*, pp. 18-26.

Kellert, S. H. (1993). In the Wake of Chaos: Unpredictable Order in Dynamical Systems. *University of Chicago Press.* , pp. 1-29.

Kim, K.-H., Youn, H.-S., & Kang, Y.-C. (2000). Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method. *IEEE Transactions on Power Systems* , pp. 559-565.

Kothari, D. P. (2008). Power system engineering - Chapter 5 and 6. *Tata McGraw-Hill.* , pp. 171-330.

Leondes, C. T. (2001). Optimization Techniques- Neural Network Systems Techniques and Applications. *Los Angeles, California: Academic Press.* , pp. 81-113.

Li, H. L. (2014). hybrid short-term power load forecasting model based on the singular spectrum analysis and autoregressive model. *Advances in Electrical Engineering*.

Liebert W., P. K. (2014). Optimal Embeddings of Chaotic Attractors from Topological Considerations. *Europhysics Lett., 14*, pp. 521-526.

Maayan Harel, S. M.-y. (2014). Concept Drift Detection Through Resampling. *roceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. II-1009-II-1017.

Mandelbrot, B. B. (1982). The fractal geometry of nature. *W. H. Freeman and co., San Francisco*.

Medha Joshi, R. S. (2017, September). Short-term load forecasting approaches: A review. *International Journal of Recent Engineering Research and Development (IJRERD)*, pp. 09-17.

Metaxiotis, K., & A. Kagiannas, D. A. (2003). Artificial intelligence in short term electric load forecasting: A state-of-the-art survey for the researcher. *Energy Conversion and Management 44*, pp. 1525-1534.

Negnevitsky, M. (2011). Artificial Intelligence: A Guide to Intelligent Systems (3rd Edition). *Canada: Pearson Education Canada.* , pp 165-217.

Office of Electricity and Energy Planning, I. M. (2011). *Iran Energy Balance in 2010.* Teheran, Iran: Iran Ministry of Energy.

Office of Electricity and Energy Planning, I. M. (2013). *Iran Energy Balance in 2012.* Teheran, Iran: Iran Ministry of Energy.

Osórioa, G. .. (2015). Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information. *Renewable Energy Elsevier,* pp. 301-307.

Pandian SC, D. K. (2006). Fuzzy approach for short term load forecasting. *Electr Power Syst*, pp. 541–548.

Park, D. C. (1991). Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems 6.2,* pp. 442-449.

Park, J., & Y. M. Park, K. Y. (1991). Composite Modelling for Adaptive Short-Term Load Forecasting. *IEEE Transactionons on Power Systems Vol. 6*, pp. 450 - 457.

Pillkahn, U. (2008). Using trends and scenarios as tools for strategy development: shaping the future of your enterprise. *John Wiley & Sons.* , pp. 13-104.

Quilumba, F. L., Lee, W.-J., Huang, H., Wang, D. Y., & Szabados, R. L. (2015). Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities. *IEEE Transactions on Smart Grid* , pp. 911-918.

Reddy, T. G. (2008). Load forecasting by a novel technique using ANN. *ARPN journal of Engineering and applied sciences, Vol. 3*, pp. 19-25.

Rothe, J. P. (2010). Hybrid and integrated approach to short term load forecasting. *nternational Journal of Engineering Science and Technology 2.12*, pp. 7127-7132.

Seema Pal, A. K. (2015). Short-Term Load Forecasting Using Adaptive. *International Journal of Novel Research in Electrical and Mechanical Engineering*, pp. 65-71.

Simon, G. a. (2006). Lag selection for regression models using high-dimensional mutual information. *European Symposium on Artificial Neural Networks-preceedings*, pp. 395-400.

Stark, J. (1994). Nonlinear Dynamics II, III: Analysis of Time series. *Centre for Nonlinear Dynamics and its Applications*, pp. 1-30.

Suparta, W. A. (2016). Modeling of Tropospheric Delays Using ANFIS. *Springer International Publishing.* , pp. 5-18.

Tascikaraoglu, A. a. (2011). Short-term residential electric load forecasting: A compressive spatio-temporal approach. *Energy and Buildings 111*, pp. 380-392.

Thai Nguyen, Y. L. (2014). Short-Term Load Forecasting Based on Adaptive Neuro-Fuzzy Inference System. *Journal of Computers, Vol. 6, No. 11*, pp. 2267-2271.

Topalli AK, E. I. (2003). A hybrid learning for neural networks applied to short term load forecasting. . *Neurocomputing 51:*, pp. 495-500.

Tranquillo, J. (2011). MATLAB for Engineering and the Life Sciences. *Morgan & Claypool.* , pp. 9-16 and 117-119.

Unkas, H. H. (2015, January). Short-term load forecasting using fuzzy logic and ANFIS. *Neural Comput & Applic. Springer*, pp. 1355-1367.

Wack, P. (1985). X Scenarios: uncharted waters ahead. *Harvard business review.*

Wang XiangJun, M. M.-H. (2012). The Comparison of Adaptive Neuro-Fuzzy Inference System (ANFIS) with Nonlinear Regression for Estimation and Prediction. *2012 International Conference on Information Technology and e-Services, Sousse, 2012, IEEE Publications*, pp. 1-7.

Wei, Y.-M. W.-C. (2006). Progress in energy complex system. *International Journal of Global Energy Issues, 25(1/2)*, pp. 109-128.

Weron, R. (2007). Modeling and forecasting electricity loads and prices: A statistical approach. England: *John Wilwy and Sons Ltd.* , pp. 67-100.

Widmer G., K. M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts,. *Mach. Learn., Vol. 23, No. 1*, pp. 69–101.

Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science, Vol. 6*, pp. 231-362.

Wu Zhi-jun, L. J.-h. (2013). Chaos-based detection of LDoS attacks,. *The Journal of Systems and Software 86*, pp. 211-221.

Xun-yi Ren, H.-s. W. (2012). Fractal Lyapunov Exponent Based Anomaly Detection of Network Traffic. *International Journal of Advancements in Computing Technology(IJACT) Volume4, Number11*, pp. 181-187.

Yap, L. (2011, July). Takens' Embedding Theorem. *OpenStax-CNX module: m38655*, pp. 1-4.

Yuh-Jye Lee, Y.-R. Y.-C. (2013). Anomaly Detection via Online Oversampling Principal Component Analysis. *IEEE Transactions on knowledge and data engineering, Vol. 25, No. 7*, pp. 1460 - 1470.

IX. Appendices

A. Implementation Codes For Load Prediction

A.1. Mutual Information Function

```
clear
clc
close all

load('data.mat');
figure(1)
plot(data,'--o','LineWidth',2,...
'MarkerEdgeColor','R',...
'MarkerFaceColor','R',...
'MarkerSize',5)

X = data(1:end-1);
Y = data(2:end);

DataNum = size(X,1);
InputNum = size(X,2);
OutputNum = size(Y,2);

MinX = min(X);
MaxX = max(X);

MinY = min(Y);
MaxY = max(Y);


XN = Normalize_Fcn(X,MinX,MaxX);
YN = Normalize_Fcn(Y,MinY,MaxY);

seq1= round(YN);
seq2=round(XN);
% P_1_2=BivarDistrFromSequence(seq1,seq2);
% P_1=UnivarDistrFromSequence(seq1);
% P_2=UnivarDistrFromSequence(seq2);


i=0;
for dt=0:60
    i=i+1;
    P_1_2=BivarDistrFromSequence(seq1,seq2)
    mutualInfo(i)=MutualInfo(seq1,seq2,dt)
end
figure(2)
plot(0:60,mutualInfo,'b-','Linewidth',1);
set(gca,'Fontsize',12);
xlabel('Time Lag');
```

```
ylabel('Mutual Information');
grid on
```

A.2. Method of Cao

```
function [E1 E2] = cao_deneme(x,tao,mmax)
%x : time series
%tao : time delay
%mmax : maximum embedding dimension
%reference:Cao, L. (1997), ``Practical method for determining the minimum
%embedding dimension of a scalar time series'', Physcai D, 110, pp. 43-50.
%author:"Merve Kizilkaya"
N=length(x);
xx=x+rand(1,N);
for m=1:mmax
   M=N-m*tao;
   YY=psr_deneme(xx,m,tao,M);
   Y=psr_deneme(x,m,tao,M);
   for n=1:M
      y0=ones(M,1)*YY(n,:);
      y1=ones(M,1)*Y(n,:);
      distance=max(abs(YY-y0),[],2);
      distance1=max(abs(Y-y1),[],2);
      [neardis nearpos]=sort(distance);
      [neardis1 nearpos1]=sort(distance1);
      newpoint=[YY(n,:) xx(n+m*tao)];
      newneig=[YY(nearpos(2),:) xx(nearpos(2)+m*tao)];
      R1=max(abs(newpoint-newneig),[],2);

      a(n)=R1/neardis(2);
      d(n)=abs(x(n+m*tao)-x(nearpos1(2)+m*tao));
   end
   E(m)=sum(a)/M;
   Ey(m)=sum(d)/M;
end
figure
E1=0.97*E(2:end)./E(1:end-1);
E2=0.97*Ey(2:end)./Ey(1:end-1);
plot(1:length(E1),E1,'k')
hold on
plot(1:length(E2),E2)
grid on
title('embedding dimension with cao method')
legend('E1','E2',4);

function Y=psr_deneme(x,m,tao,npoint)
%Phase space reconstruction
%x : time series
%m : embedding dimension
%tao : time delay
%npoint : total number of reconstructed vectors
%Y : M x m matrix
% author:"Merve Kizilkaya"
N=length(x);
if nargin == 4
   M=npoint;
else
   M=N-(m-1)*tao;
end

Y=zeros(M,m);

for i=1:m
```

```
    Y(:,i)=x((1:M)+(i-1)*tao)';
end
```

A.3. Correlation Dimension

```
function [logCr,logr]=gencorint(x,dim,tau,logr,p,w,svd,q)
%Syntax: [logCr,logr]=gencorint(x,dim,tau,logr,p,w,svd,q)
% gencorint(Data.RMR(:,2),8,2,1,1,5,0,10)
% ____________________________________________________________
%
% Calculates the generalized Correlation Integral (Cr) of a time
% series x.
%
% logCr is the the value of log(Cr).
% logr is log(range).
% x is the time series.
% dim is the embedding dimension.
% tau is the time delay.
% p is defines the norm.
% w is the Theiler's correction.
% svd is the number of singular values taken into account.
% q is the generalization index.
%
%
% References:
%
% Grassberger P, Procaccia I (1983): Characterization of strange
% attractors. Physical Review Letters 50(5):346-349
%
% Theiler J (1986): Spurious dimension from correlation algorithms
% applied tolimited time-series data. Physical Review A 34(3):2427-
% 2432
%
% Albano A M, Muench J, Schwartz C, Mees A I, Rapp P E, (1988):
% Singular-value decomposition and the Grassberger-Procaccia algorithm.
% Physical Review A38:3017-3026
%
%
% Alexandros Leontitsis
% Department of Education
% University of Ioannina
% 45110 - Dourouti
% Ioannina
% Greece
%
% University e-mail: me00743@cc.uoi.gr
% Lifetime e-mail: leoaleq@yahoo.com
% Homepage: http://www.geocities.com/CapeCanaveral/Lab/1421
%
% June 15, 2001.

if nargin<1 | isempty(x)==1
   error('You should provide a time series.');
else
   % x must be a vector
   if min(size(x))>1
      error('Invalid time series.');
   end
   x=x(:);
   % n is the time series length
   n=length(x);
end

if nargin<2 | isempty(dim)==1
```

```
   dim=2;
else
   % dim must be either a scalar or a vector
   if min(size(dim))>1
      error('dim must be a scalar or a vector.');
   end
   % dim must be an integer
   dim=round(dim);
   % dim values must be above 0
   dim=dim(find(dim>0));
end

if nargin<3 | isempty(tau)==1
   tau=1;
else
   % tau must be either a scalar or a vector
   if min(size(tau))>1
      error('tau must be a scalar or a vector.');
   end
   % tau must be an integer
   tau=round(tau);
   % tau values must be above 0
   tau=tau(find(tau>0));
end

if nargin<4  | isempty(logr)==1
   r=(max(x)-min(x))/10*(1:20)'/20;
   logr=log10(r);
else
    % logr must be either a scalar or a vector
    if min(size(dim))>1
        error('logr must be a scalar or a vector.');
    end
    % if it is a scalar, it determines the maximum range
    if length(logr)==1
        div=logr;
        % div must be positive
        if div<=0
            error('logr must be a positive scalar or vector');
        end
        r=(max(x)-min(x))/div*(1:20)'/20;
        logr=log10(r);
    else
        logr=logr(:);
        r=10.^logr;
    end
end

if nargin<5 | isempty(p)==1
   p=inf;
else
   % p must be either a scalar or a vector
   if min(size(dim))>1
      error('p must be a scalar.');
   end
   % p values must be positive
   p=p(find(p>0));
end

if nargin<6  | isempty(w)==1
   w=1;
else
   % w must be either a scalar or a vector
   if min(size(w))>1
      error('w must be either a scalar or a vector.');
   end
```

```
  % w must be an integer
  w=round(w);
  % w must be positive
  w=w(find(w>0));
end

if nargin<7 | isempty(svd)==1
   svd=[];
else
   % svd must be a scalar or a vector
   if min(size(svd))>1
      error('svd must be either a scalar or a vector.');
   end
   % svd must be an integer
   svd=round(svd);
   % svd must be positive
   svd=svd(find(svd>0));
end

if nargin<8 | isempty(q)==1
   q=2;
else
   % q must be either a scalar or a vector
   if min(size(q))>2
      error('q must be either a scalar or a vector.');
   end
end

% Only one of dim, tau, p, w, or q should be vector
l=[length(dim),length(tau),length(p),length(w),length(svd),length(q)];
if length(find(l>1))>1
  error('Only one of dim, tau, p, w, svd, or q should be vector.');
end

m=max(l);
dim=ones(1,m).*dim;
tau=ones(1,m).*tau;
p=ones(1,m).*p;
w=ones(1,m).*w;
if isempty(svd)==0
  svd=ones(1,m).*svd;
end
q=ones(1,m).*q;

for i=1:m

  % Reconstruct the time-delay phase-space
  [Y,T]=phasespace(x,dim(i),tau(i));
  if isempty(svd)==0
    svd(i)=min(svd(i),dim(i));
    % SVD on X
    [u,s,v]=svd(Y,0);

    % Calculate the Principal Components
    pc=Y*v;

    % Reconstruct the first svd Principal Components
    Y=pc(:,1:svd(i))*v(:,1:svd(i))';
  end

  % Initialize the logCr
  Cr=zeros(size(r));

  if q(i)==2 % ...fast

     for i1=1:T-w(i)
```

```
        for i2=i1+w(i):T
            dist=norm(Y(i1,:)-Y(i2,:),p(i));
            s=find(r>dist);
            Cr(s)=Cr(s)+1;
        end
    end
    Cr=2*Cr/T/(T-1);
    logCr(:,i)=log10(Cr);

  else % slow...

    for i1=1:T
        c1=zeros(size(r));
        for i2=1:T
            if i1<i2-(w(i)-1) | i1>i2+(w(i)-1)
                dist=norm(Y(i1,:)-Y(i2,:),p(i));
                s=find(r>dist);
                c1(s)=c1(s)+1;
            end
        end
        c1=c1/(T-1);
        if q(i)~=1
            Cr=Cr+c1.^(q(i)-1);
        else
            Cr=Cr+c1;
        end
    end
    if q(i)~=1
        Cr=Cr/T;
        logCr(:,i)=log10(Cr.^(1/(q(i)-1)));
    else
        Cr=log10(Cr)/T;
        logCr(:,i)=Cr;
    end

  end


figure
plot(log10(r),log10(Cr),'b-','Linewidth',1);
set(gca,'Fontsize',12);
xlabel('Ln(r)');
ylabel('Ln(Cr)');
grid on

end
```

A.4. Lyapunov Exponents

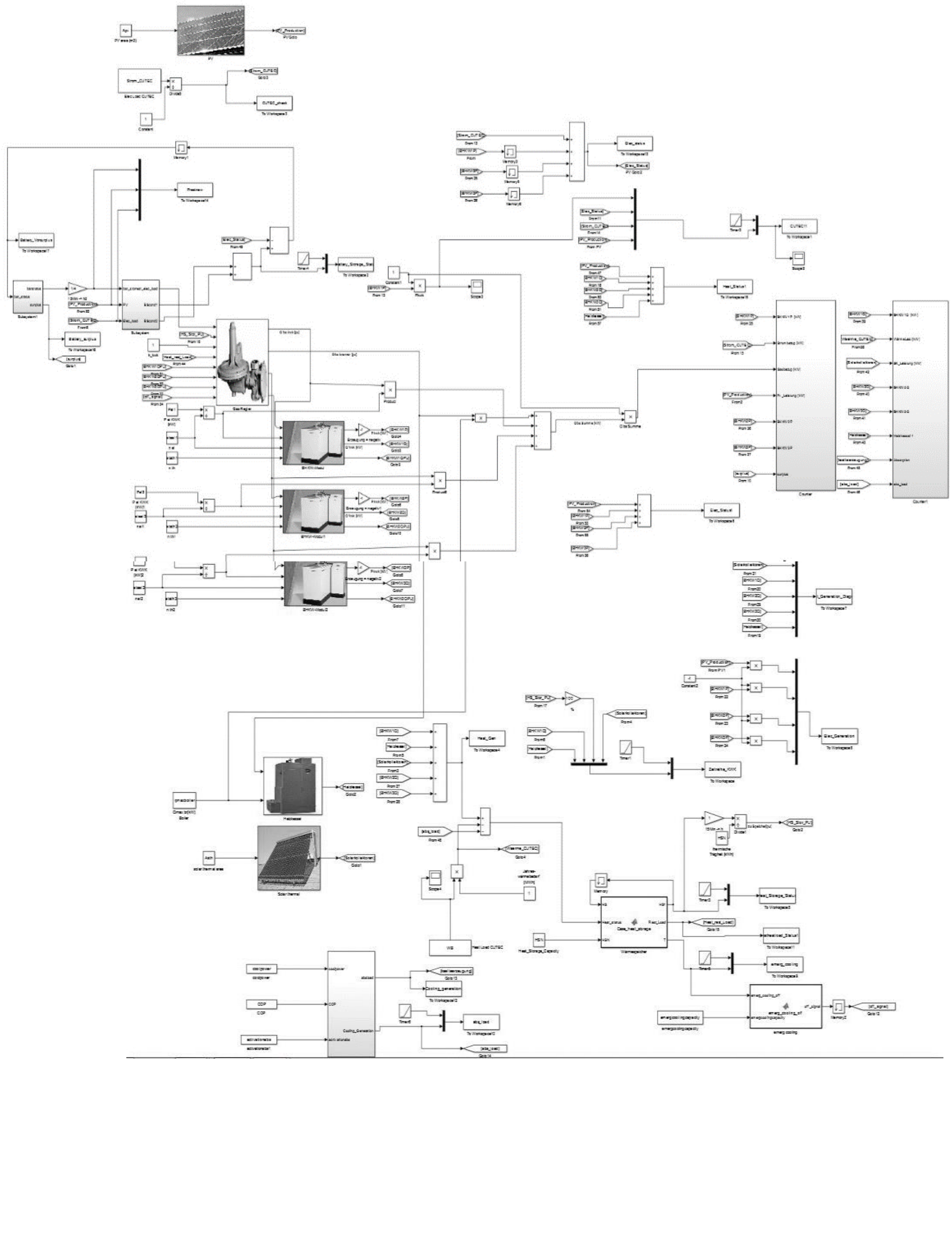
```
function d = lyarosenstein(x,m,tao,meanperiod,maxiter)
%
% lyarosenstein(Data.RMR(:,2),8,2,100,100)
% d:divergence of nearest trajectoires
% x:signal
% tao:time delay
% m:embedding dimension

N=length(x);
M=N-(m-1)*tao;
Y=psr_deneme(x,m,tao);

for i=1:M
   i
```

```
    x0=ones(M,1)*Y(i,:);
    distance=sqrt(sum((Y-x0).^2,2));
    for j=1:M
        if abs(j-i)<=meanperiod
            distance(j)=1e10;
        end
    end
    [neardis(i) nearpos(i)]=min(distance);
end

for k=1:maxiter
    k
    maxind=M-k;
    evolve=0;
    pnt=0;
    for j=1:M
        if j<=maxind && nearpos(j)<=maxind
            dist_k=sqrt(sum((Y(j+k,:)-Y(nearpos(j)+k,:)).^2,2));
             if dist_k~=0
                evolve=evolve+log(dist_k);
                pnt=pnt+1;
             end
        end
    end
    if pnt > 0
        d(k)=evolve/pnt;
    else
        d(k)=0;
    end

end
figure
plot(d,'b-','Linewidth',1);
set(gca,'Fontsize',12);
title('Prediction Error');


%% LLE Calculation
fs=2000;%sampling frequency
tlinear=15:78;
F = polyfit(tlinear,d(tlinear),1);
lle = F(1)*fs


function Y=psr_deneme(x,m,tao,npoint)
%Phase space reconstruction
%x : time series
%m : embedding dimension
%tao : time delay
%npoint : total number of reconstructed vectors
%Y : M x m matrix
% author:"Merve Kizilkaya"
N=length(x);
if nargin == 4
    M=npoint;
else
    M=N-(m-1)*tao;
end

Y=zeros(M,m);

for i=1:m
    Y(:,i)=x((1:M)+(i-1)*tao)';
end
```

B. Implementation Codes and Simulink Model for Energy Park CUTEC

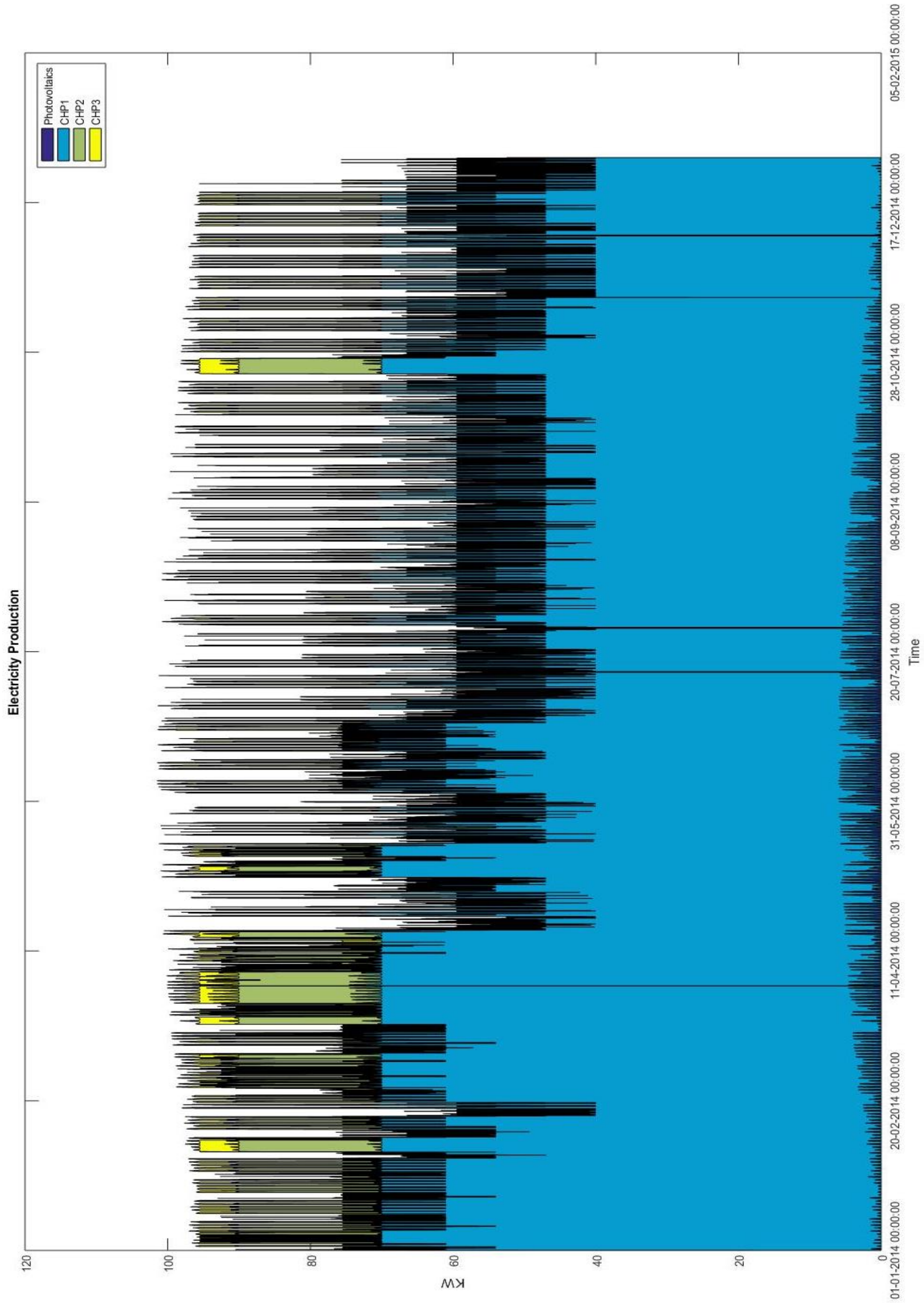
The codes related to the Energy Park includes 1500 lines, which cannot be provided here.

B.1. Simulink Model of Energy Park

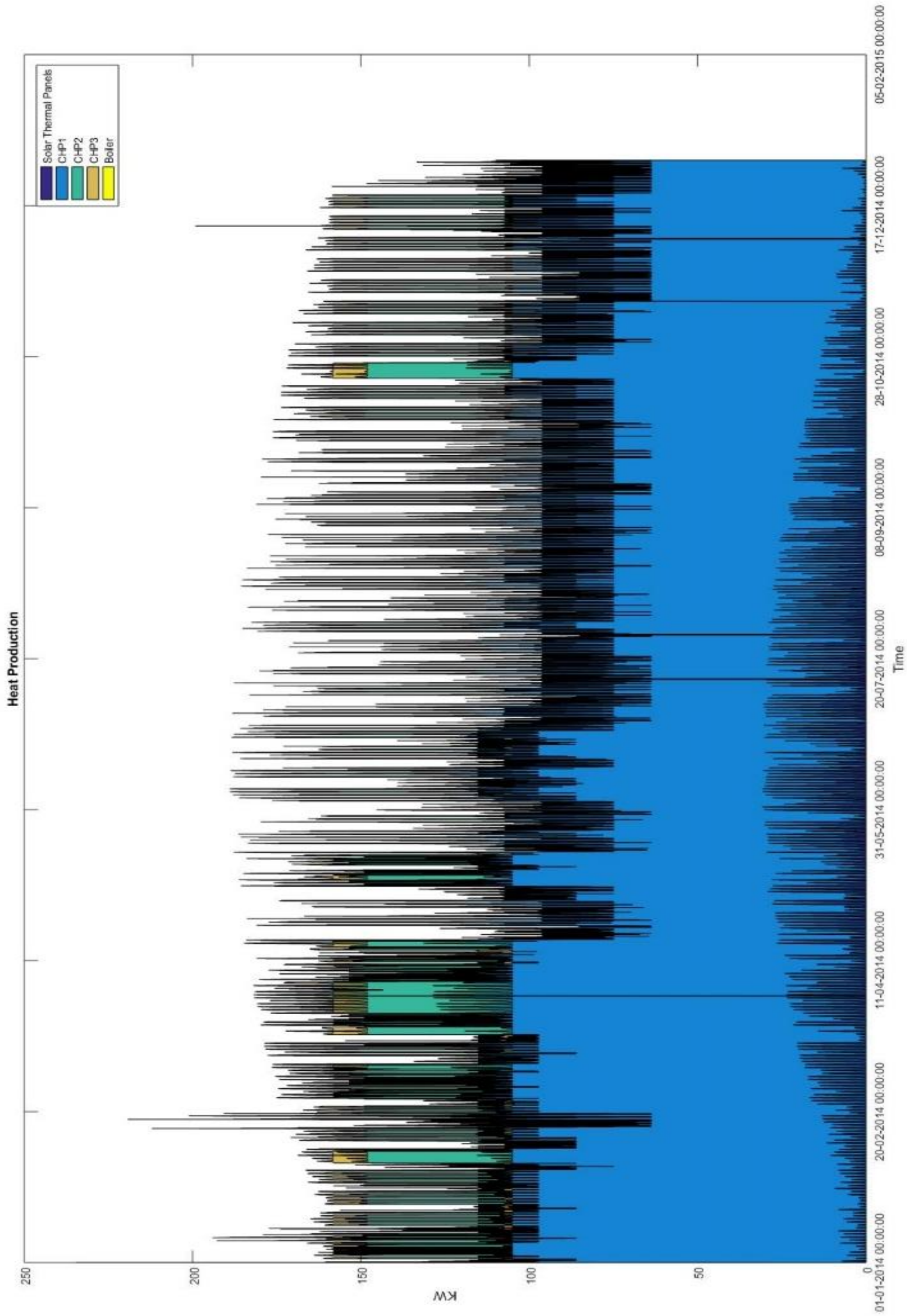


C. Simulation Results of the Energy System Model for Energy Park CUTEC in Year 2014

C.1. Electricity Production Over the Year



C.2. Heat Production Over the Year



The detailed generation of each unit in the Energy Park is available as Excel data on the electronic version of the dissertation on DVD.

D. Selected List of Presentations and Publications

2018 "New approach in Load forecasting based on Concept Drift and ANFIS", International Conference on Sustainable Energy and Environment Sensing (SEES), June 2018 , University of Cambridge , Cambridge city, United Kingdom –Oral presentation

2018 "Impact of Electromobility on the Future Standard Load Profile", 2nd international conference on Energy Economics and Energy Policy, ICEEEP 2018, Barcelona, Spain - Published in International Journal of Smart Grid and Clean Energy, vol. 8, no. 2, March 2019: pp. 164-173, ISSN: 2315-4462 (Print) (Co-Authors: M. Faulstich, J. zum Hingst, M. Jokari, S. Sangin)

2018 "The Conceptual Design of Auto-Rotary Mono-Wing Decelerators Based on Maple Seeds as an entry decent landing system for Mars explorations", European Rotorcraft Forum, September 2018, TU Delft, Netherlands – 2nd Author (Authors: S. Sangin, A. Bahri)

2018 Editorial board member in Science Research Association- Energy Journal

2018 "Naturverträgliche Energieversorgung aus 100 % erneuerbaren Energien 2050", Bundesamt für Naturschutz, BfN-Skripten 501- ISBN 978-3-89624-238-9, Bonn- (Authors: Ch. Von Haaren, J. zum Hingst, J. Wiehe, A. Walter, G. Schlömer, T. Wenzel, L. Hofmann)

2017 "Short and Long Term load forecasting using Artificial Intelligence" 12th SDEWES Conference, Dubrovnik, Croatia – Oral Presentation- Under review for publication (Co-Authors: M. Faulstich, J. zum Hingst, M. Jokari)

2016 "Case study of CCS Vs. power plant phasing out as solutions for power plant produced CO2 emission control", 11th SDEWES Conference, Lisbon, Portugal – Published in Conference Proceeding Journal (Co-Authors: M. Faulstich, J. zum Hingst, M. Jokari)

2016 "Application of heat storage in modern decentralized energy supply systems", Power to Heat 2016, Goslar, Germany- Published in the Journal: "Sektorenkopplung der Energiesysteme durch Power to Heat (Band 36)", ISBN-13: 9783736992658 (Print) (Co-Authors: W. Siemers, J. zum Hingst)

E. Curriculum Vitae

Personal Information:

Name:	Hashemifarzad
First Name:	Ali
E-Mail:	ali.hashemifarzad@tu-clausthal.de
Date of birth:	06.09.1985
Nationality	Iranian, German

Education:

2015 – 2019	Doctoral candidate at Institute of Electrical Power Engineering and Energy Systems, TU-Clausthal, Germany
2011 – 2014	Energy System Technology (Master of Science), TU-Clausthal, Germany
2004 – 2008	Power System Engineering (Bachelor of Science), Semnan National University, Iran

Working Experience:

Since 04/2015 — Clausthaler Umwelttechnik Forschungszentrum (CUTEC),

- Research Assistant
- Project Engineer
- Supervision of different lectures and Bachelor and Master theses in the fields of Energy System Technology

10/2011 – 04/2015 — Internationales Zentrum Clausthal (IZC): International Center of Clasuthal

- Student Assistant (International Applicants)

06/2011 – 07/2017 — Uni-Lotsen

- Supporting international Students as University-Guide- ***voluntary***

11/2013 – 02/2014 — Electro Padideh Alpha, Iran: Trainee

- Project Engineer
- Construction of a hybrid power plant prototype

11/2012 – 08/2013 — IPSSE - (Institute for Applied Software Systems Engineering) Goslar, Germany

- Student Research Assistant- Test Engineer

11/2006 – 10/2011	Iran Language Institute • English Teacher
11/2008 – 10/2010	Eliwell, Iran • Advising Assistant, Technical advising
07/2007 – 12/2007	Be’sat Power plant • Trainee